Pasi P. Porkka

Capacitated Timing of Mobile and Flexible Service Resources

HELSINGIN KAUPPAKORKEAKOULU
HELSINKI SCHOOL OF ECONOMICS

A-358

Pasi P. Porkka

Capacitated Timing of Mobile and Flexible Service Resources

HELSINKI SCHOOL OF ECONOMICS

ACTA UNIVERSITATIS OECONOMICAE HELSINGIENSIS

A-358

ISSN 1237-556X
ISBN 978-952-60-1000-7

E-version:
ISBN 978-952-60-1001-4

Aalto University School of Economics -
Aalto Print 2010

Author	**Pasi P. Porkka** Helsinki School of Economics Department of Business Technology
Title	Capacitated Timing of Mobile and Flexible Service Resources
Supervisors	Professor Ari P. J. Vepsäläinen Professor Olli Bräysy
Priliminary examiners	Professor Kurt Ossian Jørnsten, The Norwegian School of Economics and Business Administration, Bergen, Norway Associate Professor Ramesh Bollapragada, College of Business, San Francisco State University, USA
Chairperson of the defence	Professor Ari P. J. Vepsäläinen
Opponent	Professor Kurt Ossian Jørnsten, The Norwegian School of Economics and Business Administration, Bergen, Norway
Type of research	Based on essays
Essays	Porkka, Pasi & Vepsäläinen, Ari & Kuula, Markku (2003), Multiperiod Production Planning Carrying Over Set-up Time, *International Journal of Production Research*, Vol. 41, No. 6, pp. 1133–1148. Bräysy, Olli & Porkka, Pasi P. & Dullaert, Wout & Repoussis, Panagiotis P. & Tarantilise, Christos D. (2009), A Well-Scalable Metaheuristic For The Fleet Size And Mix Vehicle Routing Problem With Time Windows, *Expert Systems with Applications*, Volume 36, Issue 4, May 2009, pp. 8460–8475. Porkka, Pasi P. (2009), Modeling Time Capacitated Resource Allocation In Services Allowing For Split Tasks — Vehicle Routing Problem Approach. Unpublished working paper in doctoral dissertation of Porkka, Helsinki School of Economics, Finland. 46 pages. Porkka, Pasi P. (2009), Testing Of Different Time Capacitated Resource Allocation Models In Service Applications. Unpublished working paper in doctoral dissertation of Pasi P. Porkka, Helsinki School of Economics, Finland. 31 pages.

Abstract

Information revolution provides us with unforeseen opportunities for improving the productivity of services via the optimized planning of production, distribution and delivery. Now companies and clients alike can track and trace mobile resources not only inside their own factories and warehouses but also in all other service facilities and in transit between. Tracking in real time covers all products, vehicles, people and equipment. With ever shortening response times and planning periods, however, the concerns of rescheduling, rerouting, splitting and joining of production batches, product deliveries and value-added service activities will be overwhelming, especially, if realistically counting for the ramifications in time and cost of capacity each activity consumes, including all transfers and set-ups required. To be effective, this kind of time capacitated resource allocation planning also presupposes two properties from production and service resources: mobility and flexibility.

In this dissertation, we provide new views and computational methods for the real time planning of production, distribution and service delivery. The new approaches improve capacity utilization simultaneously with more flexible customer service vital for competing in the environment with increasing outsourcing and networking. Efficient capacity utilization, mobility and flexibility are achieved by the simultaneous planning of all required activities and resources by mathematical optimization applied to reliable time-based data.

Our approach to capacitated timing balances resource time used for actual production and for capacity consuming set-ups between different production batches or service activities. The explicit consideration of the capacity time consumed by all activities is critical for the realistic planning of high capacity utilization. Mobility of resources in production concerns availability in different time periods, involving costs of setting up and moving back resources through inventory build-up, work-in-process buffers and reserve machines. In service networks, mobility of resources means availability in different locations achieved by moving products, vehicles, containers and service resources, such as cleaning crews or maintenance people, and equipment, among clients, sites and geographical locations. Flexibility of resources is included by allowing production batches or service tasks to be split, joined, rescheduled and reallocated to be performed by any efficient combination of one or more different service resources, such as machines or crews.

This dissertation consists of two articles and two essays considering mobile and flexible resource allocation in time-capacitated settings. The first article deals with production planning involving shared resources and the explicit time requirements of the set-ups. The introduction of set-up carry-overs is shown to generate substantial savings in the three key factors of production costs: the number of set-ups, utilization of production capacity and level of inventory. In the second article, vehicle routing problems are solved by minimizing the sum of the traveling cost and the total cost of vehicles actually employed when transportation technology offers scale economies. New methods are introduced for efficiently solving very large problems featuring heterogenous vehicles and time windows of deliveries to as many as 1000 customers, ten times more than in earlier studies.

The two essays combine the allocation of shared resources, split tasks and variable set-ups in mobile service operations. The first essay presents a flexible service resource allocation model with a new kind of time-based splitting of work in tasks among available resources. The potential for capacity time savings achievable via this kind of modeling approach is also demonstrated by examples. In the second essay, two different time capacitated resource allocation models for service applications, one with and the other without task splitting, are tested and compared. The tests with a set of synthetic problems indicate up to 33% savings in the number of identical resources needed when the average length of tasks to split is just over half of resource capacity and the distance between task sites is short. The results imply high capacity savings potential for practical service applications by task splitting.

Despite the growing economic importance of time dependent service allocations with mobile and flexible resources, these problems have eluded the traditional modelers due to the technical and conceptual complexities involved. The new modeling and solution approaches suggested here provide some eye opening insights to the general theory while the planning methods with clearly documented results are ready for managerial applications and further development.

Key words: production planning, scheduling, service, vehicle routing, capacitated planning, resource allocation, Mixed Integer Linear Programming, optimization, heuristics, flexibility, mobility.

Acknowledgements

The focus of this dissertation evolved from production scheduling via vehicle routing and service resource allocation to a gratifying synthesis. The long journey, partly due to extensive teaching duties, has turned a blessing in disguise with crystallizing of the final results and inspiring me to ever more challenging extensions of the research. Thanks for that go to several people and institutions.

First of all, I want to thank my supervisor, Professor Ari P.J. Vepsäläinen for his unfaltering faith in my capability to really earn my doctorate and to teach at the highest of standards. Without Ari’s coaching I could have been slipping from honesty to modesty.

My deepest gratitude extends to Professor Olli Bräysy who encouraged me in my research, made me his co-researcher and introduced me to his extensive network of brilliant scientists. In research projects with Bräysy I have been witnessing the great paybacks of operations research in business and society. His example as a scholar is, and will be, a great benchmark for me.

I am indebted to Professors Kurt Jørnsten and Ramesh Bollapragada for reviewing my dissertation. Their positive feedback regenerates and energizes my desire for further deepening my knowledge in logistics and operations research. Their words of advice and perhaps joint research projects will carry me far to disseminate my results through academic journals.

My warmest appreciation goes to the faculty members and staff of the Logistics Department at the Helsinki School of Economics for sharing all these years together. Special thanks go to Mikko Tarkkala and Katri Karjalainen who never got tired of listening to my academic frustrations and cheering me up when I felt like drowning in despair.

My gratitude extends to Alfred Kordelin Foundation, the Finnish Cultural Foundation and the Foundation of HSE for financial support to my studies. Here I would also like to thank Professor Markku Kuula for his favourable letters of reference as I applied for those research grants.

Back home I am thankful to my parents Raija and Raimo Porkka and my dear friend Kaisa Auvinen for their encouragement and helping hand when social hinders, such as the almost never ending renovation of my new flat, hit the hardest.

Helsinki, January 2010

Pasi P. Porkka

PART I: Overview of the Dissertation

List of Figures

List of Tables

PART II:The Original Essays

Essay 1: Porkka, Pasi & Vepsäläinen, Ari & Kuula, Markku (2003), Multiperiod Production Planning Carrying Over Set-up Time, *International Journal of Production Research*, Vol. 41, No. 6, pp. 1133–1148.

Essay 2: Bräysy, Olli & Porkka, Pasi P. & Dullaert, Wout & Repoussis, Panagiotis P. & Tarantilise, Christos D. (2009), A Well-Scalable Metaheuristic For The Fleet Size And Mix Vehicle Routing Problem With Time Windows, *Expert Systems with Applications*, Volume 36, Issue 4, May 2009, pp. 8460–8475.

Essay 3: Porkka, Pasi P. (2009), *Modeling Time Capacitated Resource Allocation In Services Allowing For Split Tasks — Vehicle Routing Problem Approach*. Unpublished working paper in doctoral dissertation of Porkka, Helsinki School of Economics, Finland. 50 pages.

Essay 4: Porkka, Pasi P. (2009), *Testing Of Different Time Capacitated Resource Allocation Models In Service Applications*. Unpublished working paper in doctoral dissertation of Pasi P. Porkka, Helsinki School of Economics, Finland. 32 pages.

PART I:

OVERVIEV OF THE DISSERTATION

1. Introduction

The information revolution provides us with unforeseen opportunities for the real time planning of production, distribution and other services. As response times and planning periods become shorter, production batches, deliveries and services have to be rescheduled, rerouted, split and joined when simultaneously considering the capacity time each activity consumes. This kind of time capacitated resource allocation planning requires two things from production and service resources: mobility and flexibility.

This dissertation provides new views and planning methods for real time production, distribution and service planning. The new approaches improve capacity utilization simultaneously with more flexible customer service vital for surviving in ever increasing competition. Efficient capacity utilization, mobility and flexibility are achieved by the simultaneous planning of all requirements and resources by mathematical optimization applied to reliable time based data.

In production, just in time management has decreased the amount of inventories earlier used as buffers. While the ordering cycle in the seventies was weeks, the time from order to delivery in many industries has now decreased to hours. The time to respond to customer requirements has become so short that planning and changes to plans have to happen in real time. The simultaneous need for high capacity utilization and flexibility has forced companies to squeeze efficiency from their systems by advanced and more sophisticated planning methods.

In distribution, the time windows for deliveries have become very short. Deliveries have to happen on time, and often they can change at a short notice. Customers may expect fast deliveries in emergency situations but they are also often willing to pay for that additional responsiveness of service providers. A car accident may block a route and a new delivery plan has to be created in real time. Original delivery routes and even customers served by vehicles may need to be changed. The real time optimization of the routing of hundreds of vehicles serving thousands of customers in a day can only be managed with highly efficient computational methods.

In services, the interest to use advanced quantitative planning methods has started to grow slowly. The low standardization of services has made data collection and efficient planning difficult when the duration and resource consumption of tasks has been difficult to forecast. Inefficient use of capacity often stays hidden due to low standardization and the lack of systematic planning based on reliable data. Only recently the pressure for more efficiency has woken interest in the modeling and standardization of service processes. As more reliable data on service times becomes available, planning can be made more accurately, flexibly and timewise closer to the actual provision of the service. Optimizing the flexible, time capacitated allocation of service resources is then likely to become a valuable tool and approach in intensifying service capacity utilization.

In this dissertation, time capacitated modeling balances resource time used for actual production and for capacity consuming set-up times between different production batches or service activities. The explicit consideration of time related capacity consumption is critical for realistic planning with high capacity utilization. Mobility concerns inventories, work-in-process, products, vehicles and service resources, such as cleaning or maintenance staff, moving between tasks. Flexibility is included by allowing production batches or service tasks to be split, joined, rescheduled and reallocated to be performed by one or more different production resources.

This doctoral dissertation consists of four essays. The first essay on production planning extends the work of Trigeiro (1989) on capacitated lot sizing with set-up costs and the work of Sox and Gao (1999) on capacitated lot sizing with set-up costs and set-up carry-overs by including set-up time with its critical capacity consuming effects on production planning. Optimizing with set-up times and set-up carry-overs decreases the number of set-ups needed saving a substantial amount of capacity time and inventory costs compared with models not including set-up carry-overs.

The second essay on vehicle routing extends the basic Vehicle Routing Problem (VRP) to Fleet Size and Mix Vehicle Routing Problem with Time Windows (FSMVRPTW). For a general treatment of the VRP see the textbook by Toth and Vigo (2001). For a literature survey of VRP extensions see Bräysy et al. (2007a, 2007b). The Fleet Size and Mix Vehicle Routing Problem (FSMVRP) is a VRP where the homogeneous fleet assumption of the traditional VRP has been lifted. For reviews on the FSMVRP, see Salhi and Rand (1993), Osman and Salhi (1996), and Lee et al. (2008). The FSMVRPTW is a natural extension of the recently much studied Vehicle Routing Problem with Time Windows (VRPTW) surveyed by Bräysy and Gendreau (2005a, 2005b). The FSMVRPTW has been researched by Dell'Amico et al. (2007), Dondo and Cerdá (2007), Li et al. (2007), Paraskevopoulos et al. (2008). A survey on the FSMVRPTW was made by Bräysy et al. (2008). The essay in this doctoral thesis on the FSMVRPTW introduces new methods of solving very big vehicle routing problems faster and better than ever before. Another new feature with the essay problem is the vehicles cost structures exhibiting scale economies.

The third and the fourth essay extend the approach of the Split Delivery Vehicle Routing Problem (SDVRP), introduced by Dror & Trudeau (1987), where a client's demand can be served by one or more vehicles, to a time capacitated service resource allocation problem where time capacitated tasks can be worked on by one or more time capacitated multi-task resources. This kind of time capacitated routing and allocation with split tasks has not been researched in the SDVRP literature before a SDVRP survey by Archetti and Speranza (2008) or after. The third essay on time capacitated resource allocation in services presents a new one-period resource allocation model that allows split tasks between resources. The savings potential of that kind of modeling is also demonstrated. In the fourth essay two different time capacitated resource allocation models in a service application are tested and compared. Again, promising savings potential from allowing time capacitated task splitting is discovered.

Further extending this kind of time capacitated service resource allocation modeling to modeling with multiple planning periods makes it necessary to include set-up carry-overs, which will close the circle by coming back to the issues researched in the first essay.

1.1. Why Capacitated Timing of Mobile and Flexible Service Resources?

Capacitated Timing

Time is money, but much more, too. Everything happens in time and time is our most limited resource. Time once lost never comes back. Time once consumed can not be consumed again.

The production of products and services consumes time. Production capacity is constrained by time and thus inseparably linked with time. Consequently, the production rate is typically measured timewise and stated as the production of products or services per time unit. We want to use that limited time efficiently to maximize return on investment. If we can not find the most efficient way to use time, we incur an opportunity cost which is the cost difference between the most efficient resource utilization and the realized utilization.

Capacity utilization as actualized utilization divided by maximum utilization can be measured using either the amount of production or utilization time. Utilization time is the time when a production resource produces something. Often the unavoidable preparation time, set-up time, is also included in the utilization time.

In a market economy, the amount of capacity usage for production depends on demand. If the demand is less than the capacity in the long run, it is economical to produce under capacity. As the demand increases over the capacity, it may be economical to produce the most profitable product or service with full capacity. Typically, however, the capacity is not fully utilized and is used to produce different products or services. Even then, however, high capacity utilization is a goal that can be achieved either by producing more or by disposing of overcapacity.

Setting up consumes capacity time and in order to save in total set-up time, long production runs have traditionally been preferred especially in process industries where setting up may take long and cost much.

Intensified competition and customer requirements have forced companies from traditional mass production with long production runs towards more flexible production planning and shorter production runs. Doing that efficiently requires decreased set-up cost and shorter set-up times. The total time used for setting up may still have stayed the same as before. The increased number of set-ups can thus make planning for set-ups more difficult without decreasing the economic impact of set-ups on total cost and capacity utilization.

Balancing between flexibility, customer satisfaction and high capacity utilization is challenging and often requires sophisticated planning. If the planning is done efficiently, we can simultaneously increase flexibility and the increased amount of capacity by reducing the number of capacity consuming set-ups.

The capacity time consists of productive time, set-up time, possible set down time and slack. A set-up in this context is any preparation effort that is needed between two productive resource utilization periods in the production of products or services. In vehicle routing, for example, all routes between customers can be understood as unproductive set-ups. A set down is an unproductive activity that is not direct preparation for the next productive period. A set-down is, for example, a vehicle returning to a distribution center without plans considering the next tour or customers. Another example of set-down is a maintenance person who returns his van and equipment to the employer after a work day instead of driving home and starting the next day's tour directly from home with the employer's van and equipment.

In set-ups, productive time, and set-down often overlap. In vehicle routing, the traveling between clients is both setting up and actual production. The capacity time of a vehicle can be considered as fixed and the load distance cost as a variable cost. In just-in-time production, set-ups are often made off-line at the same time with the production. A manager preparing for the next day's meeting in a train on the way home combines setting up for the meeting with productive work when setting down from work. Part of setting up can also be separate from a productive resource. A worker may use equipment brought to him by another worker.

Productive time depends on the amount of produced products because the production rate is typically constant. Full capacity utilization does not create slack. The amount of slack increases as the amount of production decreases, unless the time freed from production is wasted in unnecessary set-ups. The time wasted in set-up times can be decreased by careful production planning.

As the capacity cost increases, even a small increase in available production capacity can generate a considerable increase in income. Moreover, as the utilization increases the more difficult it becomes to schedule the capacity to fulfill all requirements. As a result, the higher the capacity utilization and capacity costs, the more beneficial it is to use optimization in planning.

Time Capacitated planning and modeling are key concepts in this doctoral dissertation, and one of the most general and relevant issues in the world we live in. Actually, everything in our life is time capacitated. We think in terms of time: in terms of the present, the past and the future. All that we do is time capacitated.

The focus of time capacitated planning in this doctoral dissertation is in the planning of set-up times. In production of products or services, the use of time based modeling with focus on set-up time is justified when capacity utilization is high and it is either difficult or expensive to get additional capacity. In that case, we have to utilize the existing capacity time as efficiently as possible. By time based modeling, we can maximize the throughput that resources can produce.

When capacity utilization is high and capacity costs are low, it may be easier and less expensive to buy additional capacity when needed instead of highly optimizing the usage of existing capacity. However, if it is difficult to get additional capacity from the market, optimization can become an important way to increase the throughput of existing capacity.

When both capacity utilization and capacity cost are high, it is probably profitable to invest in optimization based planning because even a small percentage saved can mean a lot of money. On the other hand, when capacity utilization is low and the capacity cost is high, optimization may help us to identify the amount of capacity we actually need.

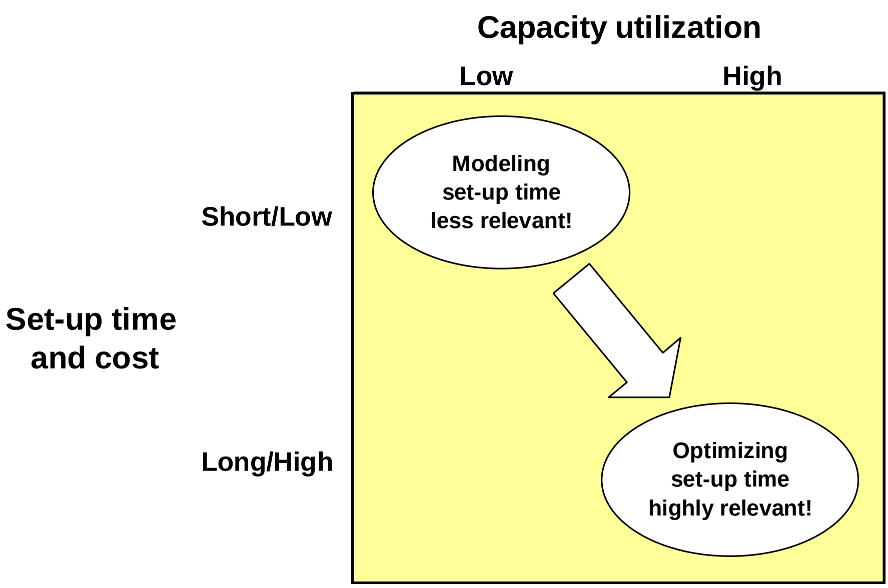

Figure 1. Optimizing set-up time is highly relevant when capacity utilization is high and set-up time is long

If set-ups are very short, we can often ignore set-up times by including set-up costs only. However, the longer the set-up times are in comparison to the production time, the more important it becomes to model set-up times, too. Figure 1 summarizes the discussion on the importance of optimizing expensive set-up time when the utilization of expensive capacity is high.

As capacity utilization is high and set-up times are long, optimizing set-up times is critical for all capacity planning because set-up time consumes capacity time and ignoring set-up times can result in plans that can not be put into practice. In those plans, production is allocated on capacity time that should be used for setting up. Figure 2 illustrates the importance of set-up optimization in relation to capacity utilization and set-up time.

Capacity utilization

Set-up time	**Low**	**High**
Short	**Set-up time can be ignored and set-ups modeled as costs only.**	**Optimizing set-up time may be relevant.** **Ignorance of set-up time may create unrealistic solutions.**
Long	**Optimizing set-up time is not critical but creates more realistic solutions than inclusion of set-up**	**Optimizing set-up time highly relevant.** **Ignorance of set-up time probably creates unrealistic solutions.**

Figure 2. Optimization of set-up time is critical when set-up times are long and capacity utilization is high.

Time is modeled in all of the four essays presented in this dissertation. The first essay models in a new, and more efficient, way a production planning problem where the allocation of set-up times determines the timing, frequency and size of production batches. The second essay models a distribution problem where traveling times between customers determine timing of customer service and the capacity of vehicles is shared between goods delivered to different customers on a delivery route. The third and the fourth essay model moving service resources whose capacity and tasks are measured as time and tasks can be flexibly worked on by one or several resources.

Time Capacitated planning assumes that capacity and efficiencies can be reliably measured and predicted as time. In production planning, production rates and set-up times are standardized. In vehicle routing traveling speed may vary to some extend because of varying traffic conditions. In services, time based planning requires the predictability of task durations meaning high standardization for service tasks.

Mobile Resources

Many resources in production and services are mobile. At a factory, the facility and the machine used for production have fixed locations, but raw materials, work-in-progress and products seldom stay long at the same place. Operators also move. In services, service resources, such as cleaning or maintenance staff, can move between customers and service tasks.

In this dissertation, the modeling of moving resources includes raw materials, work-in-progress, delivery vehicles and task performing service resources.

Flexible Resources

Resources in production and service are also flexible. A planner can decide when and where resources are used. In production planning, capacity timing for different production batches is flexible. A production batch can be split into several batches or it can be joined with other batches of the same product. Splitting, joining affects production timing that also has to be flexible.

In services, allocation of service resource time between different tasks is flexible. A performer of a task can be selected from different alternative resources, or a task can be shared by several resources either simultaneously or in a sequenced way.

1.2. The Structure of the Dissertation

This dissertation consists of four essays all considering flexible and mobile resource allocation in a time capacitated setting. Starting in production planning and dealing with shared resources and set-up carry-overs, they solve large-scale flexible distribution problems and finally combine the allocation of shared resources, split tasks and variable set-ups in mobile service operations.

The production planning essay models set-up carry-overs with set-up time instead of using set-up cost only and shows this new way of modeling improving three key production cost factors: set-ups, production capacity and inventory. The essay on vehicle routing introduces new methods of solving very big vehicle routing problems faster and better than ever before. The method simultaneously minimizes the traveling cost and the total cost of differently sized vehicles whose cost structures exhibit scale economies. The third essay presents a flexible service resource allocation model with a new kind of time based splitting of work in tasks between resources. The capacity time savings potential of that kind of modeling is also demonstrated by examples. In the fourth essay two different time capacitated resource allocation models in service applications are tested and compared. Tests indicate data structures with promising capacity time savings potential from allowing time capacitated task splitting.

All four essays bring valuable modeling and solution approaches as well as eye opening analysis to the very general and actual problem of time capacitated resource allocation. Time considerations, mobility and flexibility, which all are ever more important, have traditionally eluded modelers due to the complexities they create in models and in solving the models. In this research, they are all included in planning models with successful and clearly documented results ready for managerial applications and further research.

Chapter 2 of this introductory part of the dissertation first presents the research approach, objective and methodology of the dissertation. Then short descriptions of production lot sizing and vehicle routing are provided for readers who are not yet familiar with those concepts. Chapter 3 presents motivation and problem description, used methods, results, and contributions of authors for all four essays. Chapter 4 presents conclusions, discussion and suggestions for further research.

2. Research Approach, Objective, Methodology, and Background

This doctoral dissertation is based on four essays. The first two have already been published in refereed journals. The essays are:

Porkka, Pasi & Vepsäläinen, Ari & Kuula, Markku (2003) "Multiperiod Production Planning Carrying Over Set-up Time", *International Journal of Production Research*, Vol. 41, No. 6, pp. 1133–1148.

Bräysy, Olli & Porkka, Pasi P. & Dullaert, Wout & Repoussis, Panagiotis P. & Tarantilise, Christos D. (2009) "A Well-Scalable Metaheuristic For The Fleet Size And Mix Vehicle Routing Problem With Time Windows", *Expert Systems with Applications*, Volume 36, Issue 4, May 2009, pp. 8460–8475.

Porkka (2009), "Modeling Time Capacitated Resource Allocation In Services Allowing For Split Tasks — Vehicle Routing Problem Approach", Unpublished Working Paper in the Doctoral Dissertation of Pasi P. Porkka, Helsinki School of Economics, Helsinki, Finland.

Porkka (2009) "Testing Of Different Time Capacitated Resource Allocation Models In Service Applications", Unpublished Working Paper in the Doctoral Dissertation of Pasi P. Porkka, Helsinki School of Economics, Helsinki, Finland.

2.1. Objective

The main objective of this doctoral dissertation is to demonstrate how time capacitated modeling can provide us with more realistic and more cost efficient plans.

2.2. Research Approach and Methodology

Different methods were used in essays included in this doctoral dissertation. A common approach in all essays was that time capacitated problems were modeled, or could be modeled, as Mixed Integer Linear Programming (MILP) models. Every essay presents a new model or a new use of an existing model. Time was an important and common constraint for each model. The amount of production or services is limited by the production rate or service time, set-up time between production batches, or time spent traveling between customers.

Essays (1) and (4) compare different MILP model solutions to simulated problems using a commercial MILP solver to prove that a new modeling approach can generate more cost efficient plans than an old one. Essays (1) and (2) also provide solution heuristics. Heuristics in Essay (2) are aimed at solving large scale vehicle routing problems. Post-optimization heuristics in Essay (1) are introduced to improve solutions generated by a commercial solver. In Essays (1), (2), and (4) simulated test problems are solved by new and old solution approaches. Then solutions are compared. Essay (3) uses examples to describe and prove the savings potential of a new modeling approach. Table 1 summarizes research methodologies used in Essays (1), (2), (3), and (4).

Table 1. Research methodologies used in Essays (1), (2), (3), and (4).

	Essay 1	Essay 2	Essay 3	Essay 4
New MILP model or use of an MILP model	x	x	x	x
New heuristic solution method	x	x		
Comparison of CPLEX solutions to simulated benchmark problems	x			x
Comparison of heuristics to simulated benchmark problems		x		
Description and proving with examples			x	

2.3. Background of Essays

This subsection presents some background to the essays. First, a short introduction to lot sizing models maps the models of the first Essay (1) with other kinds of lot sizing models. Then a short overview of the Vehicle Routing Problem (VRP) is presented to give a background to Essays (2), (3), and (4). In Essay (2) the VRP is extended and solved. Essays (3) and (4) include a model quite similar to another extension of the VRP, namely the Vehicle Routing Problem with Split Deliveries (VRPSD).

A Classification of Lot Sizing Models in Production

The new models introduced in Essay (1) are extensions of the so called Capacitated Lot Sizing Problem (CLSP). Lot sizing is a wide area of research and capacitated lot sizing is only a part of it. Kuik et al. (1994) relate Capacitated Lot Sizing Problem's position to other lot sizing models by categorizing the models into two dimensions: capacity and demand. The capacity axis categorizes the lot-sizing models into capacitated and uncapacitated models. The other axis relates to the way demand is modeled: models can be distinguished on assumed knowledge of future demand. Is demand modeled as a stationary stochastic (or even constant) parameter or as a dynamic (time-depended but known) parameter? Two axes lead to the typology of lot-sizing models presented in Table 2. The models in Essay (1) assume dynamic demand and finite capacity.

Table 2. Typology of lot-sizing models (Modified from Kuik et al., 1994)

	CAPACITY	
DEMAND	**Infinite**	**Finite**
Stationary (and constant)	• Economic Order Quantity (EOQ) • Statistical Inventory Control (SIC)	• Economic Lot Scheduling Problem (ELSP) • Models based on queuing theory/Batching
Dynamic	• Multilevel Wagner-Whitin type of models	• (Multilevel) Capacitated Lot Sizing Problem = NON-carry-over problem (NCO) in essay (1) • Discrete Lot Sizing and Scheduling Problems (DLSP) • Continuous Set-up Lot sizing Problem (CSLP). • Batching / Scheduling

In the models with infinite capacity, lead-times do not depend on the operations schedule. In other words, the model's behavior concerning lead-times is independent of the level of activity, i.e., independent of the decisions taken. As lead times are considered exogenously, capacity does not play a role in decision making. In finite capacity models, on the other hand, lead-times are always considered endogenously. Capacity, be it given or to be determined, is considered as an active factor that affects lead times' length.

In a constant demand and constant capacity setting Silver et al. (1998, 443) propose the use of Economic Lot Scheduling Problem (ELSP) which creates a sequence of production that will be followed periodically. The ELSP is to find a cycle length, a production sequence, production times, and idle times, so that the production sequence can be completed in the chosen cycle, the cycle can be repeated over time, demand can be fully met, and annual inventory and set-up costs can be minimized.

Kuik et al. (1994) consider models from queuing theory, (stochastic) dynamic programming, and mixed integer linear programming as the most prominent in the area of lot-sizing. Batching analysis that makes use of models with stochastic elements, such as queuing theory, as a rule builds on the assumption of stationarity of the conditions under which the system operates: although actual conditions at time instances may vary, conditions are statistically time-invariant. Only statistical information (e.g. averages and variances) is assumed known and can be used in decision making. Regularly, the analysis yields stationary timing and sizing of batches as the best solution. For these reasons, queuing models foremost relate to analysis at the level of process choice/design in which decisions on batching (e.g. the unit size) are made on the basis of hands-off planning and control. Queuing models are capacitated models and the finiteness of processing (service) times limits the output rate of the models. Limited capacity effects become distinct when the utilization of a system approaches 100%: inventory (work-in-process) rises sharply as utilization nears 100% and the cost per unit of output rises sharply accordingly.

Deterministic models can be uncapacitated or capacitated. In contrast with queuing models, one frequently finds economies scale in uncapacitated models as the cost per unit of output decreases with the volume of output (demand). In the Capacitated Lot Sizing Problem, the scale economies of producing long batches are traded off with inventory costs. Being deterministic, mixed integer linear models are based on knowledge of values (realizations or statistics based estimations) for model parameters, such as capacity and demand, in order to determine lot sizes. These models are suitable in situations in which the state of and requirements on the system can be expressed as specific numerical values.

When demand, set-up times, or processing rates are not deterministic (or stationary), production may not follow the production plan developed by deterministic approaches considered above. If variability is high, there may be significant disruptions from using the deterministic solution in a probabilistic environment. Thus adjustments to models must be made. The Stochastic Economic Lot Scheduling Problem (SELSP) exactly parallels the ELSP with the added complexity of probabilistic (i.e. stochastic) demand, set-up times, or processing rates. Silver et al. (1998, 451 – 452) list two basic approaches to handle this problem. One is to develop a regular cyclic schedule using a solution to the deterministic problem, and then to develop a control rule that attempts to track or follow this schedule. The other is to develop a heuristic that directly and dynamically decides which product to produce next and its production quantity. The Capacitated Lot-sizing Problem is an example of such models. The dynamic models are usually finite-horizon deterministic planning models and solution procedures are implemented on a rolling horizon to take advantage of feedback on the actual inventory levels. In other words, only the first time period results of the model are implemented. At the end of every time period, new information becomes available that is used to update the model.

Capacitated Lot Sizing Problems can be subcategorized according to the way they treat set-up carry-over and set-ups. Set-up carry-overs are allowed or not allowed. Set-ups can include time, cost or both. All of the three MILP formulations studied in Essay (1) have explicitly stated set-up times. Two of them also allow set-up carry-over. Figure 3 shows the Capacitated Lot Sizing Problem categorization. The ellipses symbolize different models and the broken line is used to symbolize the models and formulations still undetected in the year 2002. The gray color highlights the model classes that had not received much attention in literature in the year 2002.

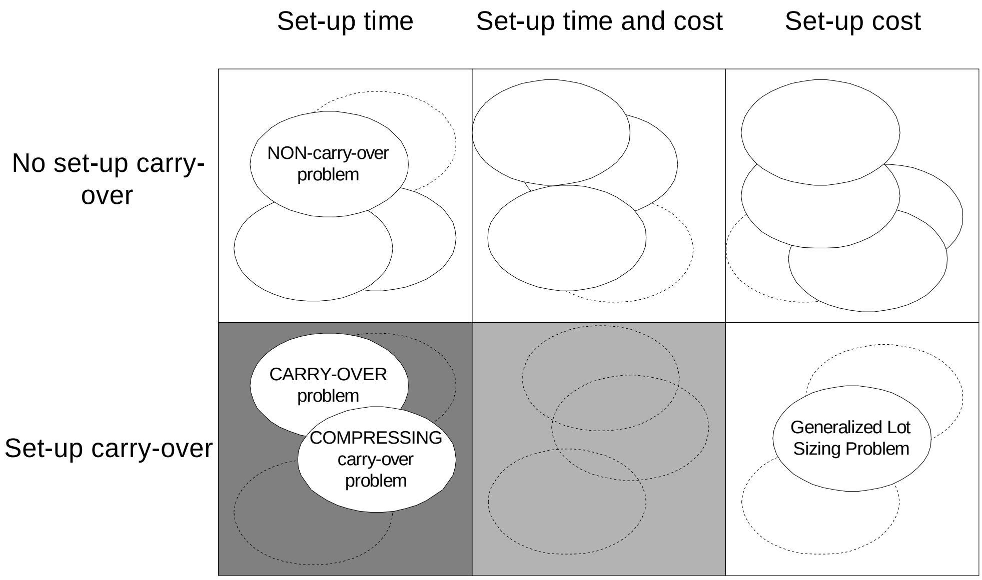

Figure 3. Classification for Capacitated Lot Sizing Problems

Vehicle Routing Problems

Transport and logistics are essential to modern Western societies. Not only do they empower individuals with unprecedented mobility, they also offer a wide variety of products and services which influence the perception of the world and even the portrayal of mankind. In general, products are either directly shipped from the supplier or manufacturer to customers or are distributed from intermediate storage points (e.g. warehouses and/or distribution centers). The latter option is highly common and gives rise to a wide variety of distribution strategies balancing risk pooling effects on inventory, inventory holding costs and transportation and distribution costs (for more information on distribution strategies see e.g. Simchi-Levi et al. 2007).

Essays (2), (3), and (4) are all associated with the Vehicle Routing Problem (VRP). The VRP concerns the distribution of goods between depots and final users. The VRP lies at the heart of these distribution problems as it addresses how the demand of customers can be satisfied at minimal cost by homogeneous vehicles located at intermediate storage facilities. The basic VRP consists of a number of geographically scattered customers, each requiring a specific weight (or volume) of goods to be delivered (or picked up). A fleet of identical vehicles dispatched from a single depot is used to deliver the goods required and once the delivery routes have been completed, the vehicles must return to the depot. Each vehicle can carry a limited weight and only one vehicle is allowed to visit each customer. It is assumed that all problem parameters, such as customer demands and travel times between customers are known with certainty.

Customers and typically one depot form a network usually modeled as a graph which can be either directed or non-directed. Typically, the transportation capacity or route length is limited leading to a situation when all customers can not be served by one route and one vehicle only. Other constraints can include periods of the day (time windows) during which customers have to be served, unloading or loading times, vehicle type, different priorities, and penalties associated with partial or total lack of service associated with customers. Routes can include deliveries, pick-ups or both. The objective is to minimize transportation costs that consist of the number of vehicles needed and actual traveling costs typically consisting of the total distance traveled.

A VRP problem is a Split Delivery Vehicle Routing Problem (SDVRP) if a client can be served by using more than one vehicle. The SDVRP is a relaxation of the classical VRP, but it still remains NP-hard. Using SDVRP instead of VRP can save in both the total distance traveled and in the number of vehicles to be used.

If every client must be serviced by exactly one vehicle, the problem is known as the Capacitated Vehicle Routing Problem (CVRP) which has been the focus of intensive research in the past 25 years.

According to Toth and Vigo (2001, 11), literature provides three different modeling approaches for VRP. In *vehicle flow formulations*, integer variables associate with each arc or edge of the graph and count the number of times the arc or edge is traversed by a vehicle. These are the frequently used models for the basic versions of VRP when the cost of the solution can be expressed as the sum of the costs associated with the arcs, and when the most relevant constraints concern the direct transition between the customers within the route, so they can be effectively modeled through an appropriate definition of the arc set and of the arc costs.

In the second form of formulations, the so-called *commodity flow formulations*, additional integer variables are associated with the arcs or edges and represent the flow of the commodities along the paths traveled by the vehicles.

In the third model type there is an exponential number of binary variables, each associated with a different feasible circuit. This VRP type is then formulated as a *Set-Partitioning Problem* (SPP) where the minimum cost collection of circuits is determined to serve each customer once and, possibly, to satisfy additional constraints.

For a general overview of the VRP, Toth and Vigo (2001) wrote a comprehensive book on Vehicle Routing Problem models and algorithms to solve them. For a literature survey of various extensions of the VRP occurring in practice, see Bräysy, Gendreau, Hasle, and Løkketangen (2007a, 2007b). A literature review on the SDVRP is presented in Essay (3).

3. Summary of Essays

3.1. ESSAY 1: Multiperiod Production Planning Carrying Over Set-Up Time

Motivation and Problem Description

Porkka (2003) is based on Porkka's Master's Thesis (2000) and was motivated by a production planning problem tackled in a company producing special papers requiring relatively long set-up times and optimal operating rates to avoid inferior quality. The company wanted to optimize production over a planning horizon consisting of 8-hours production planning periods. In this environment counting for set-up times is essential. The models found in literature in the year 2002, as the article was accepted for publication, performed unsatisfactory for the problem because they either included set-ups as fixed fees only or wasted production capacity by allocating unnecessary set-ups.

Drastic reduction in set-up times and costs in many discrete parts manufacturing processes has cut batch sizes and work in process inventories making production planning more flexible than ever. However, further research on lot sizing was still justified: Firstly, although set-up times have been reduced, they have not been eliminated. At a bottleneck facility, time wasted on set-ups always reduces the throughput of the whole system. Secondly, at the same time when shorter set-up times allow firms to reduce the manufacturing cycle at non bottleneck resources, the number of set-ups increases and the total time used for set-ups may stay the same as before. In multilevel production systems, minimizing delays and inventories between consecutive production stages may require the parallel shortening of production planning periods and set-up times which, paradoxically, may keep the time ratio between set-up times and the planning periods unchanged. For example, if we have 6-h set-ups and a 5-day (120h) production planning period or 1.2-h set-ups and a 1-day (24h) production planning period, the relative length of the set-up (5%) and thus the production planning problem remains the same. Finally, even though very small batch sizes can be reached in assembly type of manufacturing, in process industry, such as in paper production, the time and costs of frequent set-ups still force production batching and inventory holding. Because setting up does not only waste time but also consumes a lot of energy and raw materials, shortening the set-up times and the production cycles may sometimes increase the number of set-ups to the extent that the fixed fees not related to set-up time offset the yield from the increased production capacity. Thus, in systems where set-ups are of paramount importance, it is essential that they be managed explicitly.

Capacitated Lot Sizing Problems (CLSPs) are production planning models that—in a multiperiod setting—take into account the capacity constraints of a facility when determining the quantity and timing of several products over a planning horizon with known demands. The objective of CLSPs is to minimize the sum of production and inventory costs. In a single stage problem, no item can be a predecessor of another item. Costs and demand can vary over a finite horizon of discrete time periods. Backlogging is not permitted. CLSPs do not sequence or schedule jobs within a period.

Set-ups in CLSPs can be expressed as fixed fees and/or as set-up times with related costs attached. Set-ups stated as fixed fees implicitly include labor, wastage, the cost of lost production etc. Set-up times can be fixed and product specific or depend on production sequence.

In CLSP models, the inclusion of a carry-over of a set-up of a product to the next period in a case a product can be produced in subsequent periods increases solution times drastically and questions the practicality of the carry-over possibility. — In any event, no more than one production batch can be carried over between two planning periods. — Still, because setting up includes fixed costs, wastes capacity and affects inventories, manual insertion of set-up carry-overs in production plans is a common practice in many industries.

In deciding whether to include set-up carry-over possibility directly in a planning model, a production planner has to consider factors such as the average capacity utilization of a facility, the robustness of the desired production plan, the relative length of set-ups compared with the length of planning periods, and the average number of produced products during a planning period. The shorter the production planning periods and the less the average number of products produced during a planning period the bigger is the proportion of production batches with set-up carry-over potential. When set-up times are relatively short, it may be reasonable to ignore their capacity effects because the efficiency of solving CLSPs depends on the way set-ups are stated. However, during times of high capacity utilization, even the feasibility of a production planning problem may depend on the possibility to include set-up carry-overs.

Proper counting for the set-up times is crucial when the capacity consumed relative to the length of planning periods is significant. The relation often remains unchanged when production flexibility is searched by shortening both set-up times and planning periods. To efficiently decrease the number of set-ups in practice, different planning methods are applied to allow production batches, once set up, to continue over to the next planning periods.

In earlier CLSP models, the requirements of set-ups have usually been expressed in terms of cost only; the capacity consuming effect of a set-up time has been ignored. Earlier CLSP formulations also usually assume that a single set-up must be performed for an item in any period in which it is produced. However, when set-up times are considerable, setting up for a product produced last in the period $t - 1$ and first in the period t increases the total set-up cost, wastes production capacity and, especially under high capacity utilization rates, may turn a production planning problem infeasible. Even though unnecessary set-ups can be removed post-optimally and without a change in the production plan, the experiments indicate that production plans created by this method are still much more expensive than those created by models where set-up carry-overs are allowed. Figure 4 illustrates how production plans become more realistic when carry-overs and set-up time are included in a CLSP model.

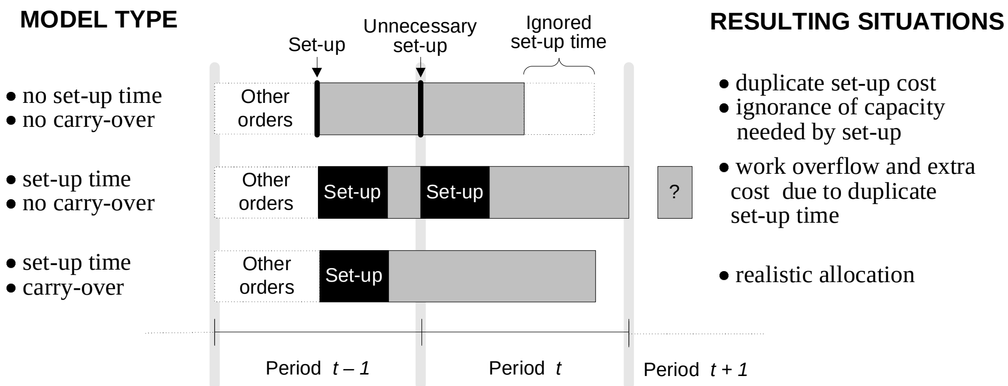

Figure 4. Three ways to allocate an order for production before the period $t + 1$ (Porkka, 2003)

The focus of this essay is on reducing the number of production capacity consuming set-ups in a continuous production environment when a product can be produced in two or more subsequent planning periods. This kind of set-up carry-over to the next period has posed problems in Mixed Integer Linear Programming (MILP) algorithms. Two different formulations of a cost minimizing carry-over model are presented and the models are compared with an earlier benchmark model without set-up carry-over. In this study set-ups are stated as set-up times with related cost to emphasize their capacity consumption effects. To avoid the complexity of separate set-up fees and costs of, possibly sequence dependent, set-up times, explicit set-up fees are excluded and set-ups costs are assumed to be directly related to the length of set-ups. Fixed and equal cost is assumed for machine time whether it is used for setting up or production.

The objective of this study is to show that, with set-up times that are relatively long in comparison to planning periods, the two new modifications of set-up carry-over models generate better production plans than the best non-carry-over benchmark model found in the literature. The study does not attempt to develop new and faster optimization algorithms to solve the carry-over problem.

Method

In the article solutions of two new set-up carry-over modeling approaches are compared to solutions generated by an existing model that does not include set-up carry-over. The non-carry-over model (NCO) is a Capacitated Lot Sizing Problem presented in Trigeiro et al. (1989). The NCO is used as a benchmark for the two new modified set-up carry-over models formulated in this study.

The first new carry-over model (CO) was formulated by adding the carry-over constraints to the NCO. CO usually schedules carry-overs between all production planning periods and allows production batches to continue over several under utilized production planning periods before a set-up for another product. In practice, this kind of under utilization pattern can either occur when a machine is run continuously, but below its capacity, or when it can be stopped and started again without a new set-up. In processes like paper production, however, the best quality may only be reached by producing batches at full or constant production rate.

The second new model, the compressing-carry-over model (CCO), forces a constant production rate of batches and allocates production stoppages between batches when capacity is underutilized. These features facilitate the production planning of certain products and give people in production planning, sales or maintenance better insight into the exact timing of planned production stoppages and unallocated capacity.

Small-scale experimental production planning problems were generated to study the effects of set-up carry-over. In addition to comparing the three optimizing models, the solutions to the NCO and the CO were post-optimally modified to imitate solutions to the CO and the CCO, in respective order. A production plan difference indicator was used to compare the allocation of production in production plans generated by different models.

Results

Solutions of an MILP based set-up carry-over models with set-up times were compared with solutions of a benchmark model without the carry-over. It was found out that the explicit counting for set-up times and carry-overs cuts down the number of set-ups and also frees a significant amount of production capacity decreasing the set-up related costs and, somewhat unexpectedly, also the inventory costs. Furthermore, experiments with heuristics that post optimally allocate carry-overs proved to capture less than one third of the cost savings generated by the MILP formulation.

Constant production rates are required in a variety of process industries. They also facilitate the planning of free capacity and production stoppages. We forced constant production rate of batches in our exact set-up carry-over model and compared the results with heuristic post optimal modification of the carry-over solutions. Again, the MILP formulation was superior to the heuristics solutions.

The significant cost savings demonstrated in our experimental results encourage the explicit inclusion of set-up carry-overs into the MILP based capacitated lot sizing models, hence motivating further research. Faster methods of solution should be developed to make carry-over models with set-up times more suitable for medium to large-scale problems with multiple machines. The future extensions of the models could include sequence dependent set-ups as well as set-up times starting at the end of one period and ending at the beginning of the next. We believe that competitive pressure to efficiently exploit production capacity motivates management to consider the carry-over of set-ups —especially the time required — as an essential feature of practical production planning.

Capacitated Lot Sizing with Set-up Carry Overs in Literature since 2003

Since the year 2002 numerous articles on capacitated lot sizing with set-up carry-over have been published including both new formulations and new solution techniques. For recent reviews on that subject see Quadt & Kuhn (2009), Jans & Degraeve (2008), Gicquel et al. (2008), Quadt & Kuhn (2008), Suerie (2006), Briskorn, D. (2006), and Suerie & Stadtler (2003).

3.2. ESSAY 2: A Well-scalable Metaheuristic for the Fleet Size and Mix Vehicle Routing Problem With Time Windows

Motivation and Problem Description

This paper addresses two of the most common extensions of the VRP occurring in practice: the presence of service time windows for customers and the use of heterogeneous vehicles. Customers often restrict the time in which they want to be serviced to a specific time interval. The resulting vehicle routing problem with time windows is probably the most studied routing problem in the literature (Bräysy & Gendreau 2005a, Bräysy & Gendreau 2005b). Because of its intrinsic complexity and practical relevance, it has been the subject of research on innovative heuristic search strategies and on solving large-scale routing problems. Extending the VRP to heterogeneous vehicles is also highly relevant because a vehicle fleet is rarely homogeneous in real-life: a fleet manager typically controls vehicles that differ in terms of equipment, carrying capacity, speed, and cost structure to better service his customers. The objective of the so-called fleet size and mix vehicle routing problem (FSMVRP) is therefore to find a fleet composition and a corresponding routing plan that minimizes the sum of routing and vehicle costs. Practical applications of FSMVRP with time windows (FSMVRPTW) are abundant and have enjoyed recent scientific attention (see e.g. Dell'Amico et al. (2007), Dondo and Cerdá (2007), Li et al. (2007), and Paraskevopoulos et al. (2008)).They are surveyed in Bräysy et al. (2008).

In spite of the large number of real customers involved, academic research on heterogeneous routing problems has been limited to relatively small problem instances. Solution approaches have often been tested on the 100-customer benchmarks of Liu and Shen (1999), derived from the well-known Solomon (1987) instances for the VRPTW. This paper focuses on the new distance-based objective variant for the FSMVRPTW, suggested in Bräysy et al. (2008)[1] and

---

[1] Refers to article Bräysy, Olli & Dullaert, Wout & Hasle, Geir & Mester, David & Gendreau, Michel (2008) "An Effective Multirestart Deterministic Annealing Metaheuristic for the Fleet Size and Mix Vehicle-Routing Problem with Time Windows", *Transportation Science*, Aug. 2008, Vol. 42 Issue 3, p371-386, 16p. The article was referred in Bräysy, Olli & Porkka, Pasi P. & Dullaert, Wout & Repoussis, Panagiotis P. & Tarantilise, Christos D. (2009), "A Well-Scalable Metaheuristic For The Fleet Size And Mix Vehicle Routing Problem With Time Windows", *Expert Systems with Applications* on page 8466 as Bräysy et al. (2008), but was mistakenly left out from article's list of references.

derive 600 new large-scale problem instances from the Gehring and Homberger (1999) problem instances for the VRPTW, using real-life data on the available vehicle types and costs. A new hybrid metaheuristic approach is described which combines the well-known Threshold Accepting and Guided Local Search metaheuristics with several search limitation strategies for a set of four local search heuristics.

Method

The proposed solution approach consists of three phases. In Phase 1 high quality initial solutions are generated by means of a limited savings heuristic. In Phase 2, the focus is on reducing the number of vehicles with a simple route elimination heuristic and in Phase 3, the Threshold Accepting (TA) (Dueck and Scheurer, 1990) and Guided Local Search (GLS) (Voudouris and Tsang, 1999) metaheuristics are used to guide a set of four local search operators to further improve the solution from Phase 2. Although the overall structure of the algorithm is similar, there are a number of major differences compared to the previous study aimed at solving large scale heterogeneous routing problems by Bräysy et al. (2008): (1) a number of algorithmic simplifications, (2) several strategies for efficiently restricting the local search and threshold accepting strategy, and (3) the introduction of a novel two-directed GLS and a simple diversification procedure.

Computational experiments were performed to examine the performance of the proposed algorithm. The computational experiments were performed using the benchmark instances proposed by Liu and Shen (1999) and 600 new benchmark instances suggested in this paper. In contrast to Liu and Shen, the sum of all vehicle costs and total distance is considered as the optimization objective, as opposed to the sum of vehicle costs and en route time. The new objective was first introduced in Bräysy et al. (2008) and it is believed to be of a higher practical value than the former objective function. The Liu and Shen benchmarks are derived from the well-known VRPTW instances of Solomon (1987). Solomon's problem sets for the VRPTW consist of 56 instances of 100 customers with randomly generated coordinates (set R), clustered coordinates (set C) or both (semiclustered RC set). The difference between Subsets R1, C1 and RC1 and R2, C2 and RC2 lie in the vehicle capacities and scheduling horizon. For each six subsets Liu and Shen introduced several vehicle types with different capacities and costs. In addition, three different vehicle cost structures A, B and C were suggested so that cost structure A refers to the largest vehicle costs and C to the smallest. To limit the computational tests, cost structure B was omitted, resulting in 112 test problems of 100 customers each. The suggested new test problems are based on the large-scale VRPTW benchmark instances of Gehring and Homberger (1999). Similar to Solomon (1987), Gehring and Homberger constructed random, clustered and semi-clustered problem sets, consisting of 200, 400, 600, 800 and 1000 customers, so that there are 60 problems of each size and 10 problems in each of the above groups. In total there are 300 problems. For the Gehring and Homberger instances, a set of vehicle types and costs were suggested. The same 8 vehicle types and costs are used for every problem size. The vehicle capacities and costs differ only between the six problem sets, as detailed in Table 3.

Table 3. Vehicle costs and capacities used for each problem set

C1		C2		R1	
Capacity	Cost	Capacity	Cost	Capacity	Cost
40	200	120	575	40	140
70	335	240	1100	70	230
100	460	350	1540	100	310
140	615	470	1975	140	405
170	715	580	2320	170	460
200	800	700	2700	200	500
240	910	820	2955	240	550
270	975	930	3160	270	565

R2		RC1		RC2	
Capacity	Cost	Capacity	Cost	Capacity	Cost
170	590	40	125	170	590
340	1115	70	205	340	1115
500	1550	100	275	500	1550
670	1945	140	355	670	1945
840	2270	170	420	840	2270
1000	2500	200	450	1000	2500
1170	2690	240	495	1170	2690
1330	2795	270	500	1330	2795

The vehicle capacities and costs in Table 3 were defined as follows. As in Liu and Shen (1999), we have used the maximum capacity in the corresponding VRPTW instance, V_B as the starting point. The cost of the vehicle with a carrying capacity V_B is the same as for the corresponding Liu and Shen (1999) 100-customer problem set. The other vehicle capacities and costs are based on real-life information collected from Finland. More precisely, we first surveyed the most typical truck types available and fixed costs related to them. Excluding vans, we found out that there are eight common vehicle types available. Liu and Shen (1999) defined only vehicle capacities smaller than the original VRPTW maximum capacity, making the problems somewhat easier to solve (optimizing the capacity utilization of larger vehicles is often harder). Here we decided also to enable vehicles with larger capacity than in the original VRPTW problem, so that we set the sixth largest vehicle carrying capacity (9 tons) to equal the V_B.

As a result, there are two truck types larger than V_B and five that are smaller in each problem. The other capacities were defined using direct relation with regard to V_B so that e.g. the capacity equaling 12 tons is obtained by multiplying V_B by 12/9 and rounding up or down with an accuracy of 5 units. The costs of other vehicle types apart from V_B were defined by first analyzing the cost relations with regard to the 9-ton vehicle. We noticed in the analysis of the real-life capacity and cost data that apart from the smallest 2–3 vehicle types, there are linear economies of scale in the cost per capacity unit. The real-life costs of the smallest vehicle types were clearly relatively more expensive than the larger ones. Therefore, they were hardly ever used in the tests done with the preliminary data. Based on this information, we decided to apply the same linear economies of scale structure over all 8 vehicle types to improve the quality of the benchmarks, by using the scaling factor defined with the three largest vehicle types. The obtained costs were then rounded to the nearest 10. This corresponds to cost structure A. As in Liu and Shen (1999), we obtained cost structure C by dividing the vehicle costs by 10, resulting in 600 test instances. Here the relationship between fixed costs and vehicle capacity is more realistic than the cost structure by Liu and

Shen (1999), where fixed costs in some cases are obviously too small in comparison to vehicle capacity.

We tested the performance of the algorithm with two different parameter settings that we denote here as Quick and Normal. The major difference between Quick and Normal was the size of the searched neighborhood and the number of improving iterations made.

Results

Test results show that strategies used in this paper are capable of significantly limiting computation time and of increasing solution quality, making them useful for solving large routing problems in general.

This paper presents a new hybrid Threshold Accepting and Guided Local Search metaheuristic that is specifically designed for solving large-scale fleet size and mix routing problems with time windows. The central part of the described algorithm consists of different strategies for balancing a limitation and intensification of the search. The computational tests were done with the benchmarks of Liu and Shen (1999) and on 600 new large-scale real-life based benchmarks suggested in this paper. A comprehensive computational study, including detailed sensitivity analysis showed that the suggested method is competitive with the previous best approach and scales almost linearly for problems up to 1000 customers.

3.3. ESSAY 3: Modeling Time Capacitated Resource Allocation in Services Allowing for Split Tasks — Vehicle Routing Problem Approach

Motivation and Problem Description

In traditional production, machines typically have fixed locations and tasks are brought to the machines as work-in-progress or raw materials. The objective in scheduling is often to maximize the utilization of the machine by feeding tasks to the machines in well-planned production batches. The machines are typically the production limiting bottleneck that is planned and scheduled carefully in order to use that bottleneck capacity as efficiently as possible.

In the modern production of products and services, a bottleneck resource may be moving instead of having a fixed location. The bottleneck resource may be moved from task to task requiring routing in addition to batching and scheduling of traditional production.

In flexible production, multi-skilled work force capable of running many different machines may be the bottleneck. Machines can have overcapacity and the production is actually limited by the working time of the workers. In many services, such as home health care, facility cleaning, waste collection and machine leasing business for construction sites, tasks may have different durations and locations, but the location of each task is fixed. Now, the moving resources are the bottleneck, and especially the available time of those resources. When planning is based on moving resources, the capacity time consists of moving between tasks, preparation for starting a task, performing the task, doing the necessary steps before leaving the task, and possibly some idle time.

Moving and allocation of moving resources is a common planning problem in many services, construction, personnel rostering and transportation planning. When planning this kind of problems on the operational level, time is an important constraint. In reality, the planned time consists of two components that are the time between tasks and the actual time needed for performing tasks. When capacity is measured as time, both of these components consume resource capacity.

In literature, however, one or both production components, production and set-ups, are typically modeled as costs only. In addition, problems are often reduced from routing AND scheduling problems into routing OR scheduling problems by fixing either of the components already in the preprocessing of the problem.

This essay studies a resource allocation problem where the time between tasks and the actual time needed for performing tasks are explicitly modeled and simultaneously solved as variables. Time is considered as key in both constraints and variables. The length of the time that a resource spends in a task is modeled as a continuous variable which allows the splitting of tasks to be performed by one or several resources. The time that each resource can work is constrained. For that kind of modeling, there is a clear gap in literature. However, it is not difficult to show that there is substantial potential for savings if flexible and time capacitated splitting of work resources between tasks is allowed. Sometimes even the feasibility of a problem may depend on whether task splitting is allowed or not.

The modeling approach in TCRAPST is very similar to the Split Delivery Vehicle Routing Problem (SDVRP). Therefore, description of the SDVRP and savings generated by split deliveries is justified. A literature research on the SDVRP is also provided to show that time capacitated task spitting has not yet been studied in SDVRP literature.

This essay has three objectives: The first objective of this essay is to get a better understanding of the capacity time saving effects of task splitting by presenting comparative examples of problems with, and without, task splitting. The second objective is to prove, by simple examples, that by task splitting between resources, up to 50% capacity time savings can be reached when compared to a situation when task splitting is not allowed. The third objective is to present a Mixed Integer Linear Programming (MILP) problem model on the Time Capacitated Resource Allocation Problem with Split Tasks (TCRAPST) and to discuss its potential applications.

Method

This essay uses small examples to demonstrate the savings potential of time capacitated modeling with flexible task splitting between resources used. Mixed Integer Linear Programming modeling is used to construct a model and to demonstrate some features of solutions. Examples of the potential business applications of the TCRAPST type of modeling are also discussed.

Results

Time Capacitated resource allocation can be applied in services where tasks are similar and resources, such as cleaning personnel, have similar and standardized skills and efficiencies. Standardization of tasks and skills makes it possible to make reliable forecasts about the capacity consumption of tasks as well as efficiencies of resources. If tasks, skills and efficiencies can not be forecast accurately enough, time capacitated resource allocation should not be used.

The essay described a resource allocation model that can be used when both capacity and requirements are expressed as time. That MILP model helps to allocate resources to perform tasks so that the number of resources gets minimized. In minimization of the number of resources needed, the routing of resources is a key because both working in tasks and moving between tasks consume capacity.

In addition to routing, task splitting was taken into account. As resources and tasks are very similar, it is often practical to let more than one resource to work in a task, especially, if a task takes a long time to complete. The MILP model presented in this essay also does that splitting of tasks between resources while simultaneously minimizing the number of resources needed in the whole system.

The focus on this essay was on the savings potential of task splitting in time capacitated modeling. Firstly, examples on that savings potential were given by comparing solutions that do not allow task splitting to solutions that include task splitting. Secondly, it was proven that task splitting can, in a theoretical case, bring up to 50% savings in comparison to a solution that does not allow task splitting. Thirdly, a Mixed Integer Linear Programming (MILP) problem model on Time Capacitated Resource Allocation Problem with Split Tasks (TCRAPST) was presented. Finally, the extensions and potential applications of the TCRAPST were discussed.

This essay concentrated on describing and proving a model that can potentially generate more efficient operational plans than the existing models. The focus was on the savings effect and, therefore, many important practical aspects, such as time windows or minimum working time constraints, were ignored. If split tasks are to be applied in practice, the model has to be extended.

Another subject for further research would be to test the savings effect with data by comparing solutions generated by the TCRAPST with solutions generated by a similar model that does not allow task splitting. Data could also be used to test the optimality of the TCRAPST solutions. Does the TCRAPST always allocate complete resource work shifts to tasks when the task length exceeds the capacity of the resource's work shift?

As an MILP formulation, the TCRAPST could only solve very small problems. If the idea of task splitting in time capacitated problems is to be put in practice, more efficient solution methods have to be developed. When developing those methods, the TCRAPST solutions, or solutions to its extensions, can be used as reference solutions to compare the quality of solutions generated by other techniques than branch-and-bound based optimization. The TCRAPST model can serve as a starting point in analyzing the implications of task splitting, set-ups, reallocations and set-downs in different industries.

Second routing application area where resource capacity can be measured in time is human resource or robot allocation and routing for different tasks.

Then a short survey is made on transportable resource allocation, routing and scheduling in cases where requirements can be measured as capacity time. Based on these surveys it becomes quite obvious that a research gap exists when it comes to modeling the flexible splitting of resource time between tasks.

3.4. ESSAY 4: Testing of Different Time Capacitated Resource Allocation Models in Service Applications

Motivation and Problem Description

In many planning and scheduling situations time is an important constraint. Time can measure the length of work in a task as well as the switching time between tasks. In production, machines with fixed locations are the resources and tasks are allocated to them. In many services, however, tasks have fixed locations and resources are allocated to tasks. For example, in house cleaning, houses have a fixed location and cleaning personnel move from house to house. The time of resources can be divided into three components that are working time, moving time and slack. If the time needed for performing a task and moving between tasks is predictable, we can route and schedule resources based on time. A typical goal in such situations is to minimize the resources needed.

In Porkka (2009a) it was shown that the time capacitated splitting of tasks to be performed by more than one resource can generate more efficient plans than when splits are not allowed. In this essay 50 simulated problems are solved to study the effects of splits in real planning situations. Problems are divided into 10 sets with task lengths and distances being generated from different probability distributions. Each problem includes 12 tasks and the purpose of the problem sets is to simulate a cluster of tasks.

Problems are first solved using the Time Capacitated Resource Allocation Problem with Split Tasks (TCRAPST) model formulated in Porkka (2009a) and then by the Time Capacitated Resource Allocation Problem (TCRAP) that solves the same problems but without splits. The TCRAP is a basic routing and allocation problem and similar models are likely to be found in literature. For this essay, the TCRAP was modeled to be a reference model to the TCRAPST when testing with simulated problems the savings effects of time capacitated task splitting. The use of the TCRAP for that purpose is new because, according to the author's knowledge, the TCRAPST was first modeled in Porkka (2009a) and this essay is the first research where its performance is compared with a reference model.

Most test problems were solved to optimality by the TCRAP but with the TCRAPST they could be solved to optimality in some cases only. Near optimal solutions were seen sufficient to demonstrate the savings generated by split tasks and to describe some interesting characteristics of solutions.

In the test problems, the resource capacity is set to be the same as the length of the planning period which makes the analysis of solutions easier. In the examples, the length of a planning period and the capacities of each resource are 8 hours which can be interpreted as the length of a working day. An easy example for a reader to keep in mind is a set of cleaning tasks in different locations and a set of workers that have to be allocated to do those tasks. The main cost in test problems is the resource time which is different for each resource. In this way, the utilization of the least expensive resources gets maximized and the utilization of the most expensive resources gets minimized.

Method

Different MILP models were compared by solving sets of simulated test problems using commercial optimization software.

Results

Two models, the TCRAPST and the TCRAP, were used to solve 10 sets of problems that simulate a service environment where task requirements are measured as capacity time needed to perform a task. Each resource had the same amount of capacity that was measured as time. Capacity is used to perform tasks, to move between tasks and to stay idle. The number of resources needed to perform tasks was minimized by simultaneously maximizing the utilization of the least expensive resources and minimizing the utilization of the most expensive resources. The TCRAPST and the TCRAP were used to generate plans that route and allocate resources for each task. When the TCRAP required each task to be completed by one resource, the TCRAPST allowed more resources to work on the same task by splitting the work load between the resources.

The study showed that the TCRAPST can generate more efficient plans than the TCRAP. Most savings appear when the average length of the tasks is just over half of the resource capacity and the average distance between the tasks is short. In such conditions the TCRAP can allocate only one resource per task and almost half of the capacity of that resource stays unused. The TCRAPST, on the other hand, can generate solutions where most resources are either fully used or completely idle.

As further research more efficient solution methods should be developed. Tests with simulated problems showed that it is impossible to find solutions to problems with realistic size by using optimization. A more efficient way to find solutions would thus be heuristic methods.

4. Discussion and Conclusions

Summary of the Results

This doctoral dissertation researched time capacitated modeling with mobile and flexible service resources in three different areas: production planning, vehicle routing and service resource allocation. In all areas, significant results were found.

The production planning Essay (1) showed that modeling with set-up times and set-up carry-overs improves three significant production cost factors: set-ups, production capacity and inventory. The results suggest the inclusion of set-up times and set-up carry-overs in real life production planning software. After the publication of Essay (1) in the year 2003, modeling with set-up carry-overs has received much attention in literature. Several new models and solution methods for practical scale problems have been published.

The vehicle routing Essay (2) introduced new methods of solving very big vehicle routing problems faster and better than ever before. Another new feature with the essay problem was the vehicles cost structures exhibiting scale economies.

Essay (3) on time capacitated resource allocation in services presented a new resource allocation model that allows split tasks between resources. Savings potential of that kind of modeling was also demonstrated.

Essay (4) tested and compared two different time capacitated resource allocation models in service application and discovered promising savings potential from allowing time capacitated task splitting.

The dissertation started with splits, continued with splits and ended with splits. Production batches were split (or joined) into batches produced at different times by the same machine. Vehicle loads were split between different customers on a distribution tour. Finally, service tasks were split between several service resources.

Limitations and Further Research

The significant cost savings demonstrated in our experimental results encourage the explicit inclusion of set-up carry-overs into the MILP based capacitated lot sizing models, hence motivating further research. Since the year 2003, extensive research on set-up times with set-up carry-overs has been published in production lot sizing, but applications of set-up carry-overs are not limited to production planning only. Set-up carry-over is a part of all capacitated resource allocation with consecutive planning periods and can be included in vehicle routing related service resource allocation. As a planning horizon is split into shorter planning periods, both set-up time and the actual production of production batches or the work on tasks can start in one period being continued in the next period. Competitive pressure to efficiently exploit production capacity motivates management to consider the carry-over of set-ups — especially the time required — as an essential feature of practical production planning and service resource allocation.

The service capacity allocation models were presented with small applications and strongly simplified to point out the significance and savings potential of flexible task splitting. To apply the approach in practice, models need to be extended to include more realistic constraints such as minimum working times, specialization, sequencing, synchronization and time windows. Some of these extensions can, in fact, simplify the problem, for example, by removing the possibility of subtours. Problem size can be decreased by decision rules in the problem generation stage but the difficulty of solving MILP formulations suggests for developing heuristic solution methods. A real case example on an organization applying task splitting should be written, too.

This doctoral dissertation focused on the planning of set-ups. A natural extension would be the inclusion of set-downs which can be defined as any unproductive activity not directly preparing for the next productive period. A set-down is, for example, a vehicle, pallet or container returning to a distribution center without plans considering the next load. Set-down in multi-period setting is also a maintenance person returning his van and equipment to the employer after a work day instead of driving home and starting the next day's tour directly from home with the employer's van and equipment.

The number, duration and costs of set-downs can be decreased by process and policy changes, but also by real time optimization and efficient rules supporting the optimization. For example, after a delivery, an empty vehicle can follow a rule to move in a direction where the next loading is most probably taking place. As demand emerges, the optimization model makes a new routing and allocation plan in real time.

As planning for splitting and joining of production batches is quite simple and mechanical to manage, there are many more decisions to make when splitting time capacitated tasks in services. Many questions should be answered: Should we optimize flexible splits or discretize the problem when pre-prosessing the data? Which tasks can be aggregated into one bigger task with subtasks sufficiently general to be worked on by many workers? Can all subtasks be allocated to any worker? Should there be a sequencing of subtasks? Which workers can work on a task at the same time? Should we have many workers working on a task simultaneously to expedite its completion? When do we know a task is ready if many workers work on it? Do workers need some additional task specific equipment? Is there some equipment that the first worker should bring and the last worker should bring back? Do resources incur any set-down costs or times after completing their work in tasks? How to handle multiple planning periods? Do we have to extend task splitting from splitting among workers to splitting among workers and planning periods? Should we split also traveling between time periods? Finding models and solutions to these problems and their combinations offers a great number of challenging questions for future research.

List of References

Archetti, C. & Speranza, M.G. (2008) "The Split Delivery Vehicle Routing Problem: A Survey", In: Golden, B., Raghavan, S. & Wasil, E.(eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges,* Operations Research/Computer Science Interfaces Series, Vol. 43, pp. 103-122, Springer, US.

Bräysy, O., Dullaert, W., Hasle, G., Mester, D. & Gendreau, M. (2008) "An Effective Multirestart Deterministic Annealing Metaheuristic for the Fleet Size and Mix Vehicle-Routing Problem with Time Windows", *Transportation Science,* Vol. 42, No. 3, pp. 371-396.

Bräysy, O. & Gendreau, M. (2005a) "Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms", *Transportation Science,* Vol. 39, No. 1, pp. 104-118.

Bräysy, O. & Gendreau, M. (2005b) "Vehicle Routing Problem with Time Windows, Part II: Metaheuristics", *Transportation Science,* Vol. 39, No. 1, pp. 119-139.

Bräysy, O., Gendreau, M., Hasle, G. & Løkketangen, A. (2007a) *A survey of heuristics for the vehicle routing problem, part I: basic problems and supply side extensions,* Working Paper, pp. 1-28, SINTEF ICT, Norway.

Bräysy, O., Gendreau, M., Hasle, G. & Løkketangen, A. (2007b) *A survey of heuristics for the vehicle routing problem, Part II: Demand Side Extensions,* Working Paper, pp. 1-34, SINTEF ICT, Norway.

Bräysy, O., Porkka, P.P., Dullaert, W., Repoussis, P.P. & Tarantilis, C.D. (2009) "A well-scalable metaheuristic for the fleet size and mix vehicle routing problem with time windows", *Expert Systems with Applications,* Vol. 36, No. 4, pp. 8460-8475.

Briskorn, D. (2006) "A note on capacitated lot sizing with setup carry over", *IIE Transactions,* Vol. 38, No. 11, pp. 1045-1047.

Dell'Amico, M., Monaci, M., Pagani, C. & Vigo, D. (2007) "Heuristic Approaches for the Fleet Size and Mix Vehicle Routing Problem with Time Windows", *Transportation Science,* Vol. 41, No. 4, pp. 516-526.

Dondo, R. & Cerdá, J. (2007) "A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows", *European Journal of Operational Research,* Vol. 176, No. 3, pp. 1478-1507.

Dror, M. & Trudeau, P. (1987) *Split Delivery Routing,* Publication No. 507 (January), Centre de recherche sur les transports, Université de Montréal, Canada.

Dueck, G. & Scheurer, T. (1990) "Threshold accepting: A general purpose optimization algorithm", *Journal of Computational Physics,* Vol. 90, No. 1, pp. 161-175.

Gehring, H. & Homberger, J.A. (1999) "Parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows", In: Miettinen, K., Mäkelä, M. & Toivanen, J. (eds.) *Proceedings of EUROGEN99,* University of Jyväskylä, pp. 57-64, Jyväskylä.

Gicquel, C., Minoux, M. & Dallery, Y. (2008) *Capacitated Lot Sizing models: a literature review,* Working Paper, pp. 1-23, Ecole Centrale Paris, Laboratoire Genie Industriel & Laboratoire d'Informatique de Paris, France.

Jans, R. & Degraeve, Z. (2008) "Modeling industrial lot sizing problems: a review", *International Journal of Production Research,* Vol. 46, No. 6, pp. 1619-1643.

Kuik, R., Salomon, M. & van Wassenhove, L.N. (1994) "Batching decisions: structure and models", *European Journal of Operational Research,* Vol. 75, No. 2, pp. 243-263.

Lee, Y., Kim, J., Kang, K. & Kim, K. (2008) "A heuristic for vehicle fleet mix problem using tabu search and set partitioning", *The Journal of the Operational Research Society,* Vol. 59, No. 6, pp. 833-841.

Li, F., Golden, B. & Wasil, E. (2007) "A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem", *Computers & Operations Research,* Vol. 34, No. 9, pp. 2734-2742.

Liu, F.H. & Shen, S.Y. (1999) "The Fleet Size and Mix Vehicle Routing Problem with Time Windows", *The Journal of the Operational Research Society,* Vol. 50, No. 7, pp. 721-732.

Osman, I.H. & Salhi, S. (1996) "Local search strategies for the vehicle fleet mix problem", In: Rayward-Smith, V.J., Osman, I.H., Reeves, C.R. & Smith, G.D.(eds.), *Modern heuristic search methods*, pp. 131-153, John Wiley & Sons, Chichester.

Paraskevopoulos, D., Repoussis, P.P., Tarantilis, C.D., Ioannou, G. & Prastacos, G. (2008) "A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows", *Journal of Heuristics,* Vol. 14, No. 5, pp. 425-455.

Porkka, P.P. (2009a) *Modeling Time Capacitated Resource Allocation In Services Allowing For Split Tasks — Vehicle Routing Problem Approach,* Unpublished Working Paper in the Doctoral Dissertation of Pasi P. Porkka, Helsinki School of Economics, Helsinki, Finland.

Porkka, P.P. (2009b) *Testing Of Different Time Capacitated Resource Allocation Models In Service Applications,* Unpublished Working Paper in the Doctoral Dissertation of Pasi P. Porkka, Helsinki School of Economics, Helsinki, Finland.

Porkka, P. (2000) *Capacitated lot sizing problem with set-up time and set-up carry-over — Case Raflatac Oy,* Master's Thesis, Helsinki School of Economics, Helsinki, Finland.

Porkka, P., Vepsäläinen, A.P.J. & Kuula, M. (2003) "Multiperiod production planning carrying over set-up time", *International Journal of Production Research,* Vol. 41, No. 6, pp. 1133-1148.

Quadt, D. & Kuhn, H. (2009) "Capacitated lot-sizing and scheduling with parallel machines, back-orders, and setup carry-over", *Naval Research Logistics,* Vol. 56, pp. 366-384.

Quadt, D. & Kuhn, H. (2008) "Capacitated lot-sizing with extensions: a review", *4OR: A Quarterly Journal of Operations Research,* Vol. 6, No. 1, pp. 61-83.

Salhi, S. & Rand, G.K. (1993) "Incorporating vehicle routing into the vehicle fleet composition problem", *European Journal of Operational Research,* Vol. 66, No. 3, pp. 313-330.

Silver, E.A., Pyke, D.F. & Peterson, R., (1998) *Inventory Management and Production Planning and Scheduling,* 3 edn., John Wiley & Sons, USA, 754 p.

Simchi-Levi, D., Kaminsky, P. & Simchi-Levi, E., (2007) *Designing and managing the supply chain. Concepts, strategies and case studies.* 3 edn., McGraw-Hill Irwin, New York, USA, 498 p.

Solomon, M.M. (1987) "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints", *Operations Research,* Vol. 35, No. 2, pp. 254-265.

Sox, C.R. & Gao, Y. (1999) "The capacitated lot sizing problem with setup carry-over", *IIE Transactions,* Vol. 31, No. 2, pp. 173-181.

Suerie, C. (2006) "Modeling of period overlapping setup times", *European Journal of Operational Research,* Vol. 174, No. 2, pp. 874-886.

Suerie, C. & Stadtler, H. (2003) "The Capacitated Lot-Sizing Problem with Linked Lot Sizes", *Management Science,* Vol. 49, No. 8, pp. 1039-1054.

Toth, P. & Vigo, D. (2001) "The vehicle routing problem", *SIAM monographs on discrete mathematics and applications,* Philadelphia, US: Society for Industrial and Applied Mathematics, USA.

Trigeiro, W.W., Thomas, L.J. & McClain, J.O. (1989) "Capacitated Lot Sizing with Setup Times", *Management Science,* Vol. 35, No. 3, pp. 353-366.

Voudouris, C. & Tsang, E. (1999) "Guided local search and its application to the traveling salesman problem", *European Journal of Operational Research,* Vol. 113, No. 2, pp. 469-499.

PART II:

ORIGINAL ARTICLES AND ESSAYS

INT. J. PROD. RES., 2003, VOL. 41, NO. 6, 1133–1148

Taylor & Francis
Taylor & Francis Group

Multiperiod production planning carrying over set-up time

P. PORKKA*, A. P. J. VEPSÄLÄINEN and M. KUULA

Set-ups eat production capacity time and continue troubling production planning, especially on bottlenecks. The shortening of production planning periods to days, shifts or even less has increased the relative length of set-up times against the periods. Yet, many production planning models either ignore set-up times or, paradoxically, split longer multiperiod batches by adding set-ups at breaks between planning periods. The MILP-based capacitated lot-sizing models that include set-up carry-overs, i.e. allow a carry-over of a set-up of a product to the next period in case a product can be produced in subsequent periods, have incorporated fixed set-up fees without consideration of capacity consumed by set-up time. Inspired by production planning in process industries where set-up times still remain substantial, we incorporated set-up time with associated cost in two modifications of carry-over models. Comparison with an earlier benchmark model without set-up carry-over shows that substantial savings can be derived from the fundamentally different production plans enforced by carry-overs. Moreover, we show that heuristic inclusion of carry-overs by removal of set-ups from non-carry-over solutions is inefficient.

1. Introduction

This study is motivated by a production planning problem tackled in a company producing special papers requiring relatively long set-up times and optimal operating rates to avoid inferior quality (Porkka 2000). The company wants to optimize production over a planning horizon consisting of 8-h production planning periods. In this environment, counting for set-up times is essential. The models found in the existing literature perform unsatisfactory for the problem because they either include set-ups as fixed fees only or waste production capacity by allocating unnecessary set-ups.

Drastic reduction in set-up times and costs in many discrete-parts manufacturing processes has cut batch sizes and work-in-process inventories making production planning more flexible than ever. However, further research on lot-sizing is still justified. First, although set-up times have been reduced, they have not been eliminated. At a bottleneck facility, time wasted on set-ups always reduces the throughput of the whole system. Second, at the same time when shorter set-up times allow firms to reduce the manufacturing cycle at non-bottleneck resources, the number of set-ups increases and the total time used for set-ups may stay the same as before. In multilevel production systems, minimizing delays and inventories between consecutive production stages may require parallel shortening of production planning periods and set-up times which, paradoxically, may keep the time ratio between set-up times and the planning periods unchanged. For example, if we have

Revision received August 2002.

† Helsinki School of Economics/Logistics, PO Box 1210, FIN-00100 Helsinki, Finland.
* To whom correspondence should be addressed. e-mail: porkka@hkkk.fi

International Journal of Production Research ISSN 0020–7543 print/ISSN 1366–588X online © 2003 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/0020754021000042995

6-h set-ups and a 5-day (120 h) production planning period or 1.2-h set-ups and a 1-day (24 h) production planning period, the relative length of the set-up (5%) and thus the production planning problem remains the same. Finally, even though very small batch sizes can be reached in assembly type of manufacturing, in a process industry, such as in paper production, the time and costs of frequent set-ups still force production batching and inventory holding. Because setting up not only wastes time, but also consumes a lot of energy and raw materials, shortening the set-up times and the production cycles may sometimes increase the number of set-ups to the extent that the fixed fees not related to set-up time offset the yield from increased production capacity. Thus, in systems where set-ups are of paramount importance, it is essential that they be managed explicitly.

Capacitated Lot Sizing Problems (CLSPs) are production planning models that—in a multiperiod setting—take into account the capacity constraints of a facility in determining the quantity and timing of several products over a planning horizon with known demands. The objective of CLSPs is to minimize the sum of production and inventory costs. In a single-stage problem no item can be a predecessor of another item. Costs and demand can vary over a finite horizon of discrete time periods. Backlogging is not permitted. CLSPs do not sequence or schedule jobs within a period.

Set-ups in CLSPs can be expressed as fixed fees and/or as set-up times with related costs attached. Set-ups stated as fixed fees implicitly include labour, wastage, cost of lost production, etc. Set-up times can be fixed and be product specific or they can depend on production sequence.

In CLSP models, inclusion of a carry-over of a set-up of a product to the next period in case a product can be produced in subsequent periods increases solution times drastically and questions the practicality of the carry-over possibility. (In any event, no more than one production batch can be carried over between two planning periods.) Still, because setting up includes fixed costs, wastes capacity and affects inventories, manual insertion of set-up carry-overs in production plans is a common practice in many industries.

In deciding whether to include a set-up carry-over possibility directly in a planning model, a production planner has to consider factors such as the average capacity utilization of a facility, the robustness of the desired production plan, the relative length of set-ups compared with the length of planning periods, and the average number of products produced during a planning period. The shorter the production planning periods and the less the average number of products produced during a planning period, the bigger the proportion of production batches with set-up carry-over potential. When set-up times are relatively short, it may be reasonable to ignore their capacity effects because the efficiency of solving CLSPs depends on the way set-ups are stated. However, during times of high capacity utilization, even the feasibility of a production planning problem may depend on the possibility of including set-up carry-overs.

The focus here is on reducing the number of production capacity consuming set-ups in a continuous production environment when a product can be produced in two or more subsequent planning periods. This kind of set-up carry-over to the next period has posed problems in Mixed Integer Linear Programming (MILP) algorithms. Two different formulations of a cost minimizing carry-over model are presented and the models are compared with an earlier benchmark model without set-up carry-over. In this study set-ups are stated as set-up times with related cost to emphasize their capacity consumption effects. To avoid the complexity of separate set-up fees and costs of, possibly sequence-dependent, set-up times, explicit set-up fees are excluded and set-ups costs are assumed to be directly related to the length of set-ups. A fixed and equal cost is assumed for machine time whether or not it is used for setting up or production.

The objective is to show that with set-up times that are relatively long in comparison with planning periods, the two new modifications of set-up carry-over models generate better production plans than the best non-carry-over benchmark model found in the literature. The study does not attempt to develop new and faster optimization algorithms to solve the carry-over problem.

Section 2 looks at earlier research on set-up carry-over. Section 3 presents the formulations of the models whose behaviour is then studied in section 4 by a set of production planning problems generated for the purpose. Section 5 presents conclusions as well as discussing the limitations of the study and prospects for future research.

2. Earlier research

In earlier CLSP models, the requirements of set-ups have usually been expressed only in terms of cost; the capacity consuming effect of set-up time has been ignored. Earlier CLSP formulations also usually assume that a single set-up must be performed for an item in any period in which it is produced (e.g. Bahl *et al.* 1987, Maes and van Wassenhove 1988, Gopalakrishnan *et al.* 1995, Haase 1996). However, when set-up times are considerable, setting up for a product produced last in the period $t-1$ and first in the period t increases the total set-up cost, wastes production capacity and, especially under high capacity utilization rates, may make a production planning problem unfeasible. Even though unnecessary set-ups can be removed post-optimally and without a change in production plan, the experiments of Sox and Gao (1999) indicate that production plans created by this method are still much more expensive than those created by models where set-up carry-overs are allowed. Figure 1 shows production plans become more realistic when carry-overs and set-up time are included into a CLSP model.

In recent years, the inclusion of set-up times and the important feature of set-up carry-overs in the CLSP have received considerable attention. Trigeiro *et al.* (1989) formulated the Capacitated Lot Sizing Problem with Set-up Time and solved it by

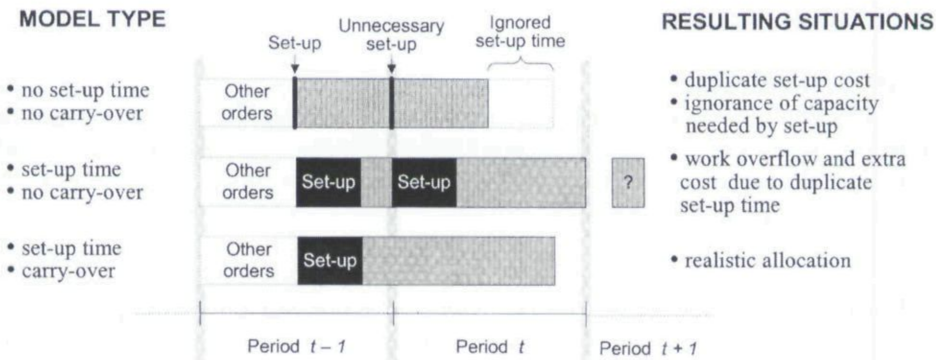

Figure 1. Three ways to allocate an order for production before the period $t+1$.

using a Lagrangean relaxation that decomposed the model into a set of uncapacitated lot-sizing problems. Diaby *et al.* (1992a, b) included both set-up costs and set-up times in their very large-scale CLSPs and solved them using a procedure based on Lagrangean relaxation and subgradient optimization. In the Continuous Set-up Lot Sizing Problem (CSLP) studied by Glassey (1968), Karmarkar and Schrage (1985) and Gascon and Leachman (1988), set-up carry-over is modelled relatively simply by allowing only one item to be produced in a given period. The Discrete Lot sizing and Scheduling Problem (DLSP) differs from the CSLP only in that the production quantity in each period is either zero or equal to the full production capacity. Drexl and Haase (1995) described a model that extends the CSLP and the DLSP by allowing at most two different items to be produced in each period and develops a backward-oriented, regret-based random sampling method for generating solutions. Haase (1996) modelled the CLSP with set-up carry-over restricted to at most one period and presented a parametrized priority rule based backward-oriented heuristic for solving the problem. The heuristic performs a local search on the parameter space and returns the best solutions. Gopalakrishnan *et al.* (1995) formulated the CLSP with set-up carry-over and a similar multi-machine, multi-family problem. Their model assumes all items to have the same set-up cost and time. Sox and Gao (1999) presented a more efficient Generalized Lot Sizing Problem (GLSP) that extends the CLSP by allowing set-up carry-overs, and the CSLP by allowing the production of multiple items in a single period. Sox and Gao, reformulated the GLSP to a network problem. They also formulated a restricted version of the model, where a production run can carry over to at most one period, and developed a Lagrangean decomposition heuristic for it. Both models closely approximated GLSP's optimal solutions but they could be solved much faster than the basic MILP formulation.

Sox and Gao compared the behaviour of four models including the modified CLSP and the GLSP. In the modified CLSP, the problem was first solved as a CLSP. Then set-up costs were minimized by introducing set-up carry-overs, where possible, while still using the optimal CLSP lot sizes. Although the cost saving effect of set-up carry-overs in the modified CLSP was significant, the procedure was still worse than the GLSP, which gains additional cost reduction by simultaneously optimizing holding and set-up costs while allowing set-up carry-over.

In parallel with our study, Gopalakrishnan *et al.* (2001) developed and tested a Tabu-search heuristic for the CLSP with set-up carry-over and set-up time but they did not present concise MILP formulation of the problem. The near-optimal solutions of their test procedure support the conclusions of our study.

3. Three studied models

In this study, the solutions to three small-scale capacitated lot-sizing problem models are compared by varying capacity utilization rates and set-up times. In the three models set-ups are expressed as set-up time with time-related cost, but fixed set-up fees can easily be included if needed.

3.1. *Non-carry-over model*

The first model, the non-carry-over model (NCO), is a CLSP (Trigeiro *et al.* 1989) and is based on the work of Billington (1983) and Billington *et al.* (1983). The NCO is used as a benchmark for the two new modified set-up carry-over models formulated in this study. The NCO is formulated as follows.

$\nu \in (1, \ldots, V)$	set of products,
$t \in (1, \ldots, T)$	set of periods in the planning horizon,
$I_{\nu t}$	inventory of ν on hand at the end of t,
$x_{\nu t}$	amount of production for ν in t,
$z_{\nu t}$	1, if set-up is incurred for ν in t; 0 otherwise,
A_ν	amount of capacity consumed per one unit of ν,
C_t	amount of capacity available per period,
$D_{\nu t}$	demand for ν in t,
$I_{\nu 0}$	beginning inventory of ν,
M	positive number at least as big as the total demand for all products over the planning horizon ($\sum_{t=1}^{T} \sum_{\nu=1}^{V} D_{\nu t}$),
S_ν	set-up time for ν,
c	cost of used capacity unit,
$h_{\nu t}$	holding cost per unit of ν on hand at the end of t.

$$\min \quad \sum_{t=1}^{T} \sum_{\nu=1}^{V} [c(z_{\nu t} S_\nu + x_{\nu t} A_\nu) + I_{\nu t} h_{\nu t}] \tag{1}$$

$$\text{s.t.} \quad \sum_{\nu=1}^{V} (z_{\nu t} S_\nu + x_{\nu t} A_\nu) \leq C_t, \qquad \forall\, t, \tag{2}$$

$$x_{\nu t} \leq z_{\nu t} M, \qquad \forall\, \nu \text{ and } t, \tag{3}$$

$$x_{\nu t} + I_{\nu,t-1} - I_{\nu t} = D_{\nu t}, \qquad \forall\, \nu \text{ and } t, \tag{4}$$

$$I_{\nu t}, x_{\nu t} \geq 0, \qquad \forall\, \nu \text{ and } t,$$

$$z_{\nu t} \in \{0, 1\}, \qquad \forall\, \nu \text{ and } t.$$

The objective function (1) minimizes the sum of production and holding costs. The constraints include capacity limits (2), the constraint requiring set-ups in periods when production occurs (3) and the material balance equation (4).

3.2. *Carry-over model*

Porkka (2000) formulates the carry-over model (CO) by replacing or adding the carry-over constraints from the GLSP (Sox and Gao 1999) to the NCO. Constraint (3) is replaced by constraint (5), and constraints (6–8) are added as follows.

$\zeta_{\nu t}$	1, if a set-up carry-over for ν is incurred from period $t-1$ to period t; 0, otherwise,
$\zeta_{\nu 1}$	1, if ν is in production at the same time as the production planning model is run; 0, otherwise.

$$x_{\nu t} \leq (z_{\nu t} + \zeta_{\nu t}) M, \qquad \forall\, \nu \text{ and } t, \tag{5}$$

$$\sum_{\nu=1}^{V} \zeta_{\nu t} \leq 1, \qquad \forall\, t \geq 1, \tag{6}$$

$$\zeta_{\nu t} - z_{\nu,t-1} - \zeta_{\nu,t-1} \leq 0, \qquad \forall\, \nu \text{ and } t \geq 1, \tag{7}$$

$$\zeta_{\nu t} + \zeta_{\nu,t-1} - z_{\nu,t-1} + z_{j,t-1} \leq 2, \quad \forall \nu, t \geq 2, \text{and } j \neq \nu \tag{8}$$

$$\zeta_{\nu t} \in \{0, 1\}, \quad \forall \nu \text{ and } t.$$

Constraint (5) requires that either $z_{\nu t} = 1$ or $\zeta_{\nu t} = 1$ whenever $x_{\nu t} > 0$. Constraint (6) requires that production of no more than one product carries over from $t-1$ to t. Constraint (7) forces $\zeta_{\nu t} = 0$ if both $z_{\nu,t-1}$ and $\zeta_{\nu,t-1} = 0$, i.e. the facility did not run process ν during period $t-1$. However, if $\zeta_{\nu t} = 1$ then either $z_{\nu,t-1} = 1$ or $\zeta_{\nu,t-1} = 1$ or both. Constraint (8) is the multiperiod synchronization constraint for processes that continue over more than two periods. If $\zeta_{\nu t} = \zeta_{\nu,t-1} = 1$, then if any process j (other than ν) is run during $t-1$, ν must be set up again to be carried over to t.

3.3. *Compressing carry-over model*

The CO usually schedules carry-overs between all production planning periods and allows production batches to continue over several under utilizated production planning periods before a set-up for another product. In practice, this kind of under utilization pattern can either occur when a machine is run continuously, but below its capacity, or when it can be stopped and started again without a new set-up. In processes like paper production, however, the best quality may only be reached by producing batches at full or constant production rate.

The compressing-carry-over model (CCO) by Porkka (2000) forces constant production rates of batches and allocates production stoppages between batches when capacity is under utilized. These features facilitate the production planning of certain products and gives people in production planning, sales or maintenance better insight into the exact timing of planned production stoppages and unallocated capacity. The CCO is formulated by replacing constraint (2) in CO with constraint (9) and by introducing the new constraints (10) and (11):

u_t amount of idle production capacity units in t,
ε_t a very small number $[< 1/(1 + C_t), \forall t]$.

$$u_t + \sum_{\nu=1}^{V} (z_{\nu t} S_\nu + x_{\nu t} A_\nu) = C_t, \quad \forall t, \tag{9}$$

$$\zeta_{\nu,t-1} + \zeta_{\nu t} - z_{\nu,t-1} + \epsilon_{t-1} u_{t-1} \leq 2, \quad \forall t \text{ and } \nu, \tag{10}$$

$$\epsilon_{t-1} \zeta_{\nu t} \leq C_{t-1}, \quad \forall t \text{ and } \nu, \tag{11}$$

$$u_t \geq 0, \quad \forall t.$$

Constraint (9) defines the amount of idle time in each production planning period by restricting it to the difference between available production capacity and used production capacity. Constraint (10) prevents the continuation of a process ν from period $t-2$ to t if the period $t-1$ includes idle time.

Constraint (11) completes constraint (10) by preventing set-up carry-overs between periods $t-1$ and t when the period $t-1$ does not have any production capacity.

4. Comparison of model variants

Small-scale experimental production planning problems were generated to study the effects of set-up carry-over. In addition to comparing the three optimizing models, the solutions to the NCO and the CO were post-optimally modified to imitate solutions to the CO and the CCO, in respective order. A production plan difference indicator was used to compare the allocation of production in production plans generated by different models.

4.1. *Problem generation*

The experimental production planning problems were generated as follows:

Problems size of three products and 10 production planning periods with equal capacity $[C_t = C_{t+1}\ \forall t \in (1, \ldots, T-1)]$ were considered sufficient to demonstrate the differences between solutions. *Demands* for all of the 10 periods were generated from U[0, 100] probability distribution. One unit of *capacity per unit of production* was used for all products in test problems. *Set-up times* within problems were set equal for all products. The *beginning inventories of the first period* were arbitrarily set to 50 units for all products. *Set-up cost/holding cost ratio* 200 determined the cost of capacity used for setting up for a product in comparison with the cost of holding a unit of the same product in stock over a planning period. With this ratio and without capacity constraint, the Economic Order Quantity formula would suggest production interval of just below three periods and, thus with the planning horizon of 10 periods, three to four set-ups for each item. To keep the model generator simple *first period set-up carry-over variables* were fixed to zero.

Four different sets of problems were generated according to their *relative length of set-up time* in comparison with planning period's length. Twelve problem types were created as combinations of relative set-up times [2, 5, 10, 15%] and lot-for-lot capacity utilization rates [75, 100, 110%]. Ten production planning problems of each of the 12 problem types were generated by separate simulations and solved by each of the three tested models to create 360 different production plans.

Based on average demand and set-up times, *production capacities per production planning periods* were calculated for target average lot-for-lot capacity utilization rates 75, 100 and 110%. The presence of set-up time in capacitated lot-sizing problems allows feasible solutions to problems that in lot-for-lot lot sizing need more than 100% capacity. The average *capacity requirements per period* were calculated by using the following.

Let

- N number of products ν,
- D_ν average demand for product ν per period,
- c $C_t =$ final average production capacity per period,
- P_ν amount of production for product ν produced in a capacity unit,
- S_ν set-up time for product ν,
- U target average capacity utilization rate:

$$c = \frac{\sum_{\nu=1}^{N}\left(\frac{D_\nu}{P_\nu} + S_\nu\right)}{U} \Leftrightarrow U = \frac{\sum_{\nu=1}^{N}\left(\frac{D_\nu}{P_\nu} + S_\nu\right)}{c}.$$

In each problem type the generation of production planning problems was continued until there were 10 problems that were feasible for all of the three studied optimization models. Regeneration of some problems caused an unavoidable and unmeasurable bias in problem generation.

4.2. *Post-optimal modification of solutions*

Additional binary variables required by carry-over models make them slower to solve than the NCO model. However, the solution time of capacitated lot-sizing problems does not relate polynomially to the number of binary variables: the relative increase in solution time tends to exceed the relative increase in the number of binary variables, but it cannot be known whether the effect of, for example, doubling the number of binary variables is to ten-, 100-, or 1000-fold the solution time.

Shortening the solution times would make capacitated lot-sizing problems more attractive for practitioners. To save time when still planning with set-up carry-overs, the NCO solutions were post-optimally modified by two slightly different heuristics (NCO-H1 and NCO-H2). The heuristics use capacity consuming set-up times with associated costs when the optimization based modified CLSP by Sox and Gao (1999) included only set-up costs. To combine the batches capable of making up a set-up carry-over, the heuristics remove the 'degenerate' set-ups using the following steps. (1) Identify carry-over opportunities between successive set-ups for the same product. (2) Prevent two successive set-up carry-overs for a product if other products are produced between the suggested carry-overs. (3) Prevent the suboptimal and carry-over wasting allocation illustrated in table 1. (4) Allow only one carry-over per period. (Because the set-up times and the related costs in the experimental problems are equal for each product and period, there is no need to prioritize between the set-ups after steps 1–3.) Table 2 gives an example of the substitution of set-ups (SU) for carry-overs (CO) in a NCO solution.

The first heuristic (NCO-H1) calculates the objective values as the NCO solution deducted by the capacity cost associated with unnecessary set-ups. The second heuristic (NCO-H2), in addition to removing the unnecessary set-ups, also moves part of the production in carry-over batches backwards in time to fill in the production breaks between batches otherwise caused by the removed set-ups.

Because the solution times of the CCO appeared to be much longer than those of the CO model, we also experimented with the post-optimal inclusion of batch compression into the CO solutions. The heuristic (CO-H) developed for this purpose forces carry-over production batches to be produced at full capacity from their beginning to their end. In batches with one carry-over, the production is moved from its second production period backwards so that all unallocated production capacity in its first production period becomes completely used. The same idea is

	Suboptimal		
	$t-1$	t	$t+1$
Product ν	SU	**CO**	SU
Product $j \neq \nu$	SU	SU	
	Optimal		
	$t-1$	t	$t+1$
Product ν	SU	SU	**CO**
Product $j \neq \nu$	SU	**CO**	

Table 1. Suboptimal and optimal way to allocate the set-up carry-overs.

Set-ups in a non-carry-over solution										
	Per 1	Per 2	Per 3	Per 4	Per 5	Per 6	Per 7	Per 8	Per 9	Per10
Product 1	SU	SU		SU	SU	SU			SU	
Product 2	SU	SU		SU			SU	SU		
Product 3		SU	SU			SU	SU	SU	SU	SU

Modified set-ups, carry-overs and carry-over batches										
	Per 1	Per 2	Per 3	Per 4	Per 5	Per 6	Per 7	Per 8	Per 9	Per10
Product 1	SU	SU		SU	CO	CO			SU	
Product 2	SU	CO		SU			SU	CO		
Product 3		SU	CO			SU	CO	SU	CO	SU

Table 2. Example of the set-up adaptation heuristic for the NCO solutions.

applied to batches with two carry-overs, the longest batches that occurred in the CO solutions to the generated problems of this study.

4.3. *Indicators of production plan differences*

The following statistical measure by Sox and Gao (1999) was applied to evaluate the difference between the lot sizes generated by the three different models for identical lot-for-lot lot sizing problems:

$$\Delta(x, x') = \frac{\frac{1}{2}\sum_{t=1}^{T}\sum_{\nu=1}^{N}|x_{\nu t} - x'_{\nu t}|}{\sum_{t=1}^{T}\sum_{\nu=1}^{N} d_{\nu t}},$$

where x and x' are the vectors of optimal lot-sizes for compared models. The function Δ computes the fraction of total demand for all items that the first model plans to produce in different periods than the second model. The $\Delta(x, x')$ values in solution difference indicators demonstrate the difference in the lot sizes generated by different models in different problem instances.

4.4. *Experimental results*

To compare the solution to the three problems, the solution data within each problem type was averaged to attain (3 models* 12 problem types =) 36 average solutions for each variable of interest. In this section the word ‘solutions’ refers to these average solutions to the 12 problem types.

- *Objective values:* as table 3 shows, the inclusion of set-up carry-overs to the NCO decreased costs between 1.3 and 14.8% but adding the batch compression to the CO increased the costs between 0.1 and 3.6%. Keeping the lot-for-lot capacity utilization rate constant and increasing the relative length of set-up times increased savings from carry-overs: eliminating a longer set-up frees more production capacity. Savings seem to increase when lot-for-lot capacity utilization rate decreases: with increasing capacity utilization and the number of set-ups being approximately the same, relatively more of the capacity is allocated to production and relatively less for set-ups. As a consequence, the higher the capacity utilization the less is the relative contribution of a set-up time on the whole capacity consumption.

Relative set-up time		2%			5%			10%			15%		
Lot-for-lot utilization rate		75%	100%	110%	75%	100%	110%	75%	100%	110%	75%	100%	110%
Optimizing models	NCO	68 760	87 299	90 012	25 668	34 358	35 894	10 681	16 311	17 134	6 503	9 787	10 563
	CO	67 762	86 081	88 878	24 639	33 292	34 779	9 739	15 298	15 998	5 544	8 718	9 506
	CCO	67 915	86 123	88 934	24 835	33 363	34 853	9 952	15 416	16 110	5 753	8 924	9 677
Post-optimal modification heuristics	NCO-H1	68 577	86 698	89 549	25 546	34 053	35 598	10 640	16 098	16 849	6 508	9 668	10 367
	NCO-H2	68 590	86 727	89 566	25 562	34 070	35 612	10 653	16 115	16 860	6 508	9 715	10 382
	CO-H	68 551	86 503	89 154	25 496	33 890	35 295	10 530	16 044	16 619	6 410	9 595	10 263
Percentual differences		1.48	1.39	1.26	4.01	3.10	3.11	8.82	6.21	6.63	14.75	10.92	10.01
	(NCO-CO)/NCO	1.26	1.35	1.20	3.29	2.90	2.90	6.82	5.49	5.97	11.54	8.81	8.39
	(CCO-CO)/CCO	0.22	0.05	0.06	0.75	0.21	0.21	2.15	0.77	0.70	3.63	2.31	1.77
	(CCO-CO)/NCO	0.22	0.05	0.06	0.72	0.21	0.21	2.00	0.72	0.65	3.21	2.11	1.62
	(NCO—NCO-H1)/NCO	0.3	0.7	0.5	0.5	0.9	0.8	0.4	1.3	1.7	−0.1	1.2	1.9
	(NCO—NCO-H2)/NCO	0.3	0.7	0.5	0.4	0.8	0.8	0.3	1.2	1.6	−0.1	0.7	1.7
	(C-H—CCO)/C-H	0.9	0.4	0.2	2.6	1.6	1.3	5.5	3.9	3.1	10.3	7.0	5.7

Table 3. Average objective function values and the same percentual differences between them.

- *Solution times:* the emphasis of this study is to compare and describe the behaviour of different models in terms of objective values and production plans. Because solution times vary much depending on hardware, only the relative solution times of the three different models were compared. On average, the CO used about 8.5 times and the CCO about 16.8 times more time to solve a production planning problem than the NCO. The result is interesting because due to its additional constraints, the CCO could be expected to have a smaller feasible region and thus shorter solving times than the CO.

  The relative length of set-up time does not seem to have any clear effect on solution times. However, keeping the relative set-up times constant, solution times appear to correlate negatively with lot-for-lot capacity utilization. Some stand-alone experiments indicated that changes in other problem parameters, such as the set-up/holding cost ratio, could radically affect solution times.
- *Set-ups and set-up carry-overs*: the CO allocates no costs on set-up carry-overs. Nor does it require 100% capacity utilization for set-up carry-overs to take place. Consequently, the CO allocates set-up carry-overs between all production planning periods having allocated demand and capacity. On the other hand, the CCO, by forcing full production rate for all production batches, generates production stoppages that continue between production planning periods reducing the number of set-up carry-overs.

  In all of the 12 problem types, the NCO on average allocated most and the CO least set-ups in production plans. The number of set-ups in the NCO solutions was about the same as what the Economic Order Quantity would have suggested. The number of set-ups in the CO solutions was on average 30% and in the CCO solutions 22% less than that. The number of set-ups in the first few periods is to some extent biased by the initial inventories fixed to the average demand per period. Table 4 presents the average number of set-ups generated by different models.
- *Capacity utilization*: as table 5 indicates, depending on the problem type and the model, the three capacitated lot sizing models needed from 9.9 to 51.5% less capacity than the lot-for-lot production planning. Keeping the lot-for-lot capacity utilization constant and increasing the relative length of set-up times seems to lead to increased capacity savings.
- *Inventories*: without set-up time, set-up carry-overs yield lower production costs but not lower inventory costs. In contrast, when the set-up time with associated cost is included, set-up carry-overs reduce both production and inventory costs by freeing capacity and in that way allowing later start of production.

  As can be seen from tables 4 and 6, inventory levels and the related costs generated by the three capacitated lot-sizing problem models seem to decrease as the number of set-ups decreases. The NCO solutions include the biggest inventories and the CCO solutions the smallest. However, the study was probably too small to guarantee for the CCO solutions the smallest average inventories in every problem type: in four of the 12 problem types, the average inventories in CCO solutions were bigger than in the CO solutions.
- *Carry-over inclusion heuristics*: the modified NCO solutions created by the NCO-H1 need fewer set-ups and, consequently, have lower costs than the original NCO solutions. However, as can be seen from table 3, the decrease

Relative set-up time	2%			5%			10%			15%		
Lot-for-lot utilization rate	75%	100%	110%	75%	100%	110%	75%	100%	110%	75%	100%	110%
NCO	10.6	14.6	13.70	10.9	12.5	12.9	9.7	11.3	12.0	9.2	9.8	11.1
CO	8.0	9.5	8.40	7.7	8.7	8.4	7.4	8.0	8.2	6.9	7.1	7.4
CCO	8.6	9.6	8.70	9.0	9.0	8.4	8.5	9.3	8.8	8.5	8.7	9.0
NCO-H1	9.6	11.6	11.40	10.3	11.0	11.4	9.5	10.2	10.6	9.2	9.2	10.1

Table 4. Average number of set-ups produced by different models.

Relative set-up time	2%			5%			10%			15%		
Lot-for-lot utilization rate	75%	100%	110%	75%	100%	110%	75%	100%	110%	75%	100%	110%
NCO	67.6	86.3	89.1	61.3	83.6	87.2	48.1	75.9	80.4	39.9	63.9	71.5
CO	67.0	85.3	88.1	59.7	81.8	84.9	45.8	72.6	76.6	36.4	59.9	65.9
CCO	67.2	85.4	88.1	60.5	81.9	84.9	46.9	73.9	77.2	39.0	62.1	68.3
NCO-H1	67.4	85.9	88.7	61.1	82.7	86.4	47.9	74.8	79.1	40.0	63.1	70.0
NCO-H2	67.4	85.9	88.7	61.1	82.7	86.4	47.9	74.8	79.1	40.0	63.1	70.0
CO-H	67.1	85.4	88.1	59.8	81.5	84.9	45.8	72.6	76.7	36.6	59.9	65.9
(LFL cap.ut–NCO cap.ut)/LFL cap.ut	**9.9**	13.7	19.0	18.3	16.4	20.7	35.9	24.1	26.9	46.8	36.1	35.0
(LFL cap.ut–CO cap.ut)/LFL cap.ut	10.7	14.7	19.9	20.4	18.2	22.8	38.9	27.4	30.4	**51.5**	40.1	40.1
(LFL cap.ut–CCO cap.ut)/LFL cap.ut	10.4	14.6	19.9	19.3	18.1	22.8	37.5	26.1	29.8	48.0	37.9	37.9
(LFL cap.ut–NCO-H1 cap.ut)/LFL cap.ut	10.1	14.1	19.3	18,6	17.3	21.4	36.2	25.2	28.1	46.7	36.9	36.4
(LFL cap.ut–NCO-H2 cap.ut)/LFL cap.ut	10.1	14.1	19.3	18,6	17.3	21.4	36.2	25.2	28.1	46.7	36.9	36.4
(LFL cap.ut–CO-H cap.ut)/LFL cap.ut	10.5	14.6	19.9	20.3	18.5	22.8	39.0	27.4	30.3	51.3	40.1	40.1

Data are percentage.

Table 5. Comparison of capacity utilizations of the NCO, the CO and the CCO.

Relative set-up time		2%			5%			10%			15%		
Lot-for-lot utilization rate		75%	100%	110%	75%	100%	110%	75%	100%	110%	75%	100%	110%
NCO	Set-up	2 120	2 920	2 740	2 180	2 500	2 580	1 940	2 260	2 400	1 840	1 960	2 220
	Production	65 500	83 541	86 435	22 371	30 873	32 294	7 675	12 912	13 703	3 495	6 573	7 312
	Inventory	1 161	838	837	1 117	985	1 020	1 066	1 139	1 031	1 168	1 254	1 031
	Total	68 781	87 299	90 012	25 668	34 358	35 894	10 681	16 311	17 134	6 503	9 787	10 563
	Set-up (%)	3	3	3	8	7	7	18	14	14	28	20	21
	Production (%)	95	96	96	87	90	90	72	79	80	54	67	69
	Inventory (%)	2	1	1	4	3	3	10	7	6	18	13	10
	Total (%)	100	100	100	100	100	100	100	100	100	100	100	100
CO	Set-up	1 600	1 900	1 680	1 540	1 740	1 680	1 480	1 600	1 640	1 380	1 420	1 480
	Production	65 000	83 541	86 435	22 371	30 873	32 294	7 675	12 912	13 703	3 495	6 573	7 312
	Inventory	663	641	762	728	679	805	584	786	655	669	725	714
	Total	67 763	86 082	88 877	24 639	33 292	34 779	9 739	15 298	15 998	5 544	8 718	9 506
	Set-up (%)	2	2	2	6	5	5	15	10	10	25	16	16
	Production (%)	97	97	97	91	93	93	79	84	86	63	75	77
	Inventory (%)	1	1	1	3	2	2	6	5	4	12	8	8
	Total (%)	100	100	100	100	100	100	100	100	100	100	100	100
CCO	Set-up	1 720	1 920	1 740	1 800	1 800	1 680	1 700	1 860	1 760	1 700	1 740	1 800
	Production	65 500	83 541	86 435	22 371	30 873	32 294	7 675	12 912	13 703	3 495	6 573	7 312
	Inventory	695	662	758	653	690	880	578	644	647	558	611	565
	Total	67 915	86 123	88 933	24 824	33 363	34 854	9 953	15 416	16 110	5 753	8 924	9 677
	Set-up (%)	3	2	2	7	5	5	17	12	11	30	19	19
	Production (%)	96	97	97	90	93	93	77	84	85	61	74	76
	Inventory (%)	1	1	1	3	2	3	6	4	4	10	7	6
	Total (%)	100	100	100	100	100	100	100	100	100	100	100	100
Set-up costs (%)	(NCO–CO)/NCO	25	35	39	29	30	35	24	29	32	25	28	33
	(NCO–CCO)/NCO	19	34	36	17	28	35	12	18	27	8	11	19
	(CO–CCO)/CO	−8	−1	−4	−17	−3	0	−15	−16	−7	−23	−23	−22
Inventory costs (%)	(NCO–CO)/NCO	43	24	9	35	31	21	45	31	36	43	42	31
	(NCO–CCO)/NCO	40	21	9	42	30	14	46	43	37	52	51	45
	(CO–CCO)/CO	−5	−3	1	10	−2	−9	1	18	1	17	16	21

Table 6. Division of coss between set-up, production and inventory costs in the solutions to different models.

	NCO	CO	CCO	NC-H2	CO-H
NCO	100	47	46	1	41
CO	47	100	21	47	37
CCO	46	21	100	47	41
NCO-H2	1	47	47	100	41
CO-H	41	37	41	41	100

Data are percentages.

Table 7. Average solution difference indicator values for production pland generated by different models.

in objective function values was less than one-third of the savings that could have been gained by using the CO model. The NCO-H2 solutions had the same number of set-ups as the NCO-H1 solutions but due to backward movements of production, higher inventory levels leading to higher objective function values.

- *Batch compression heuristic*: the experimental results indicate that it is more expensive to include post-optimally batch compression into the CO solutions than to use the CCO model. The total costs of the CCO solutions were from 0.2 to 10.3% lower than the total costs of the modified CO solutions where batch compression was post-optimally included by using the CO-H heuristic. Keeping other factors constant, the cost differences between the CO-H solutions and the CCO solutions increased when relative set-up time increased and decreased when lot-for-lot capacity utilization rate increased.
- *Solution difference indicator*: according to table 7, the production plans for the same production planning problems clearly vary depending on the used model. The relative set-up times and the lot-for-lot capacities do not seem to have systematic effects on production plan differences.

  The CO and CCO reallocate about 50% of the production when compared with the NCO solutions. On the other hand, there is hardly any change in production plans when carry-overs are post-optimally included into the NCO solutions using the NCO-H2 that after the elimination of the unnecessary set-ups moves part of the production backwards in time to keep the production batches continuous. Correspondingly, the CO-H solutions differ more from the CCO solutions than from the original CO solutions. The formulation of the set-up carry-over and the batch compression features into the NCO and the CO changes their solutions radically enough to demonstrate that those changes can not be efficiently imitated by the post-optimal heuristics of this study.

5. Conclusions

Proper counting for the set-up times is crucial when the capacity consumed relative to the length of planning periods is significant. The relation often remains unchanged when production flexibility is searched by shortening both set-up times and planning periods. To decrease efficiently the number of set-ups in practice, different planning methods are applied to allow production batches, once set up, to continue over to next planning periods.

In this paper, we modify an existing MILP-based set-up carry-over model by using set-up times instead of fixed set-up fees and compare its behaviour with a benchmark model without the carry-over. The explicit counting for set-up times and carry-overs cuts down the number of set-ups and also frees a significant amount of production capacity decreasing the set-up related costs and, somewhat unexpectedly, also the inventory costs. We also experimented with heuristics that post-optimally allocate carry-overs, but this approach proved to capture less than one-third of the cost savings generated by the MILP formulation.

Constant production rates are required in a variety of process industries. They also facilitate the planning of free capacity and production stoppages. We forced constant production rates of batches in our exact set-up carry-over model and compared the results with heuristic post-optimal modification of the carry-over solutions. Again, the MILP formulation was superior to the heuristics solutions.

The significant cost savings demonstrated in our experimental results encourage the explicit inclusion of set-up carry-overs into the MILP-based capacitated lot-sizing models, hence also motivating some further research. Faster methods for a solution should be developed to make carry-over models with set-up times more suitable for medium to large-scale problems with multiple machines. Future extensions of the models could include sequence-dependent set-ups as well as set-up times starting in the end of one period and ending at the beginning the next. We believe that competitive pressure to exploit efficiently production capacity motivates management to consider the carry-over of set-ups—especially the time required—as an essential feature of practical production planning.

Acknowledgement

This study was supported by a grant for scientific research from the Alfred Kordelinin foundation in Finland.

References

Bahl, H. C., Ritzman, L. P. and Gupta, J. N. D., 1987, Determining lot sizes and resource requirements: a review. *Operations Research*, **35**, 329–345.

Billington, P. J., 1983, Multi-level lot-sizing with a bottleneck work center. PhD dissertation, Graduate School of Business and Public Administration, Cornell University.

Billington, P. J., McClain, J. O. and Thomas, L. J., 1983, Mathematical approaches to capacity-constrained MRP systems: review, formulation and problem reduction. *Management Science*, **29**, 1126–1141.

Diaby, M., Bahl, H. C., Karwan, M. H. and Zionts, S., 1992a, A lagrangean relaxation approach for very-large-scale capacitated lot-sizing. *Management Science*, **38**, 1329–1340.

Diaby, M., Bahl, H. C., Karwan, M. H. and Zionts, S., 1992b, Capacitated lot-sizing and scheduling by Lagrangean relaxation. *European Journal of Operational Research*, **38**, 444–458.

Gascon, A. and Leachman, R. C., 1988, A dynamic programing solution to the dynamic, multi-item, single-machine scheduling problem. *Operations Research*, **36**, 50–56.

Glassey, C. R., 1968, Minimum change-over scheduling of several products on one machine. *Operations Research*, **16**, 343–352.

Gopalakrishnan, M., Ding, K., Bourjolly, J. and Mohan, S., 2001, A Tabu-search heuristic for the capacitated lot-sizing problem with set-up carryover. *Management Science*, **47**, 851–863.

Gopalakrishnan, M., Miller, D. M. and Schmidt, C. P., 1995, A framework for modelling set-up carry-over in the capacitated lot sizing problem. *International Journal of Production Research*, **33**, 1973–1988.

Haase, K. 1996, Capacitated lot sizing with linked production quantities of adjacent periods. Working Paper 334, Institut für Betriebswirschaftslehre, Christian-Albrechts-Universität zu Kiel, Germany.

Karmarkar, U. S. and Schrage, L., 1985, The deterministic dynamic product cycling problem. *Operations Research,* **33**, 326–345.

Maes, J. and Van Wassenhove, L., 1988, Multi-item, single-level capacitated dynamic lot-sizing heuristics: a general review. *Journal of the Operational Research Society* **39**, 991–1004.

Porkka, P., 2000, Capacitated lot sizing problem with set-up time and set-up carry-over—Case Raflatac Oy. Master's thesis, Helsinki School of Economics, Finland.

Sox, C. R. and Gao, Y., 1999, The capacitated lot sizing problem with set-up carry-over. *IIE Transactions*, **31**, 173–181.

Trigeiro, W. W., Thomas, L. J. and McClain, J. O., 1989, Capacitated lot sizing with setup times. *Management Science,* **35**, 353–366.

Expert Systems with Applications 36 (2009) 8460–8475

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A well-scalable metaheuristic for the fleet size and mix vehicle routing problem with time windows

Olli Bräysy [a], Pasi P. Porkka [b], Wout Dullaert [c,d,*], Panagiotis P. Repoussis [e], Christos D. Tarantilis [e]

[a] Agora Innoroad Laboratory, Agora Center, P.O. Box 35, FI-40014 University of Jyväskylä, Finland
[b] Helsinki School of Economics, Logistics, P.O. Box 1210, FIN-00101 Helsinki, Finland
[c] Institute of Transport and Maritime Management Antwerp, University of Antwerp, Keizerstraat 64, B-2000 Antwerp, Belgium
[d] Antwerp Maritime Academy, Noordkasteel Oost 6, 2030 Antwerpen, Belgium
[e] Athens University of Economics and Business, Evelpidon 47A and Leukados 33,11362, Athens, Greece

ARTICLE INFO

Keywords:
Vehicle routing
Heterogeneous vehicles
Metaheuristics

ABSTRACT

This paper presents an efficient and well-scalable metaheuristic for fleet size and mix vehicle routing with time windows. The suggested solution method combines the strengths of well-known threshold accepting and guided local search metaheuristics to guide a set of four local search heuristics. The computational tests were done using the benchmarks of [Liu, F.-H., & Shen, S.-Y. (1999). The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research Society, 50*(7), 721–732] and 600 new benchmark problems suggested in this paper. The results indicate that the suggested method is competitive and scales almost linearly up to instances with 1000 customers.

1. Introduction

Transport and logistics are essential to modern Western societies. Not only do they empower individuals with unprecedented mobility, they also offer a wide variety of products and services which influence perception of the world and even portrayal of mankind. In general, products are either directly shipped from the supplier or manufacturer to customers or are distributed from intermediate storage points (e.g. warehouses and/or distribution centers). The latter option is highly common and gives rise to a wide variety of distribution strategies balancing risk pooling effects in inventory, inventory holding costs and transportation and distribution costs (for more information on distribution strategies see e.g. Simchi-Levi, Kaminsky, & Simchi-Levi, 2008).

The vehicle routing problem (VRP) lies at the heart of these distribution problems as it addresses how the demand of customers can be satisfied at minimal cost by homogeneous vehicles located at intermediate storage facilities. The basic VRP consists of a number of geographically scattered customers, each requiring a specified weight (or volume) of goods to be delivered (or picked up). A fleet of identical vehicles dispatched from a single depot is used to deliver the goods required and once the delivery routes have been completed, the vehicles must return to the depot. Each vehicle can carry a limited weight and only one vehicle is allowed to visit each customer. It is assumed that all problem parameters, such as customer demands and travel times between customers are known with certainty. For a general overview of the VRP, we refer to the textbook by Toth and Vigo (2001). For a literature survey of various extensions of the VRP occurring in practice, we refer to Bräysy, Gendreau, Hasle, and Løkketangen (2007a, 2007b).

This paper addresses two of the most common extensions of the VRP occurring in practice: the presence of service time windows for customers and the use of heterogeneous vehicles. Customers often restrict the time in which they want to be serviced to a specific time interval. The resulting vehicle routing problem with time windows is probably the most studied routing problem in the literature (Bräysy & Gendreau, 2005a, 2005b). Because of its intrinsic complexity and practical relevance, it has been the subject of research on innovative heuristic search strategies and on solving large-scale routing problems. Extending the VRP to heterogeneous vehicles is also highly relevant because a vehicle fleet is rarely homogeneous in real-life: a fleet manager typically controls vehicles that differ in terms of equipment, carrying capacity, speed, and cost structure to better service his customers. The objective of the so-called fleet size and mix vehicle routing problem (FSMVRP) is therefore to find a fleet composition and a corresponding routing plan that minimizes the sum of routing and vehicle costs. Practical applications of FSMVRP with time windows (FSMVRPTW) are abundant and have enjoyed recent scientific attention (Dell'Amico, Monaci, Pagani, & Vigo, 2006; Dondo & Cerdá, 2007; Li, Golden, & Wasil, 2007; Paraskevopoulos, Repoussis, Tarantilis, Ioannou, & Prastacos, 2007). They are surveyed in Bräysy et al. (2007).

* Corresponding author. Address: Institute of Transport and Maritime Management Antwerp, University of Antwerp, Keizerstraat 64, B-2000 Antwerp, Belgium. Tel.: +32 3 275 51 41; fax: +32 3 275 51 50.
E-mail addresses: Olli.Braysy@jyu.fi (O. Bräysy), porkka@hse.fi (P.P. Porkka), wout.dullaert@ua.ac.be (W. Dullaert).

0957-4174/$ - see front matter
doi:10.1016/j.eswa.2008.10.040

In spite of the large number of real customers involved, academic research on heterogeneous routing problems has been limited to relatively small problem instances. Solution approaches have often been tested on the 100-customer benchmarks of Liu and Shen (1999), derived from the well-known Solomon (1987) instances for the VRPTW. In this paper we focus on the new distance-based objective variant for the FSMVRPTW, suggested in Bräysy et al. (2007) and derive 600 new large-scale problem instances from the Gehring and Homberger (1999) problem instances for the VRPTW, using real-life data on the available vehicle types and costs. A new hybrid metaheuristic approach is described which combines the well-known threshold accepting and guided local search metaheuristics with several search limitation strategies for a set of four local search heuristics.

The remainder of the paper is structured as follows: In Section 2, we describe the algorithm that is a modification of the method of Bräysy et al. (2007), specifically designed for solving large-scale problems. The results of the computational experiments are given in Section 3, including both comprehensive sensitivity analysis and comparison to previous work. Section 4 concludes the paper.

2. The algorithm description

The proposed solution approach consists of three phases. In Phase 1 high quality initial solutions are generated by means of a limited savings heuristic. (see sub Section 2.1.). In Phase 2 the focus is on reducing the number of vehicles with a simple route elimination heuristic (see sub Section 2.2) and in Phase 3, the threshold accepting (TA) (Dueck & Scheurer, 1990) and guided local search (GLS) (Voudouris & Tsang, 1998) metaheuristics are used to guide a set of four local search operators to further improve the solution from Phase 2 (see Section 2.3.). Although the overall structure of the algorithm is similar, there are a number of major differences compared to the previous study aimed at solving large scale heterogeneous routing problems by Bräysy et al. (2007): a number of algorithmic simplifications, several strategies for efficiently restricting the local search and threshold accepting strategy, and the introduction of a novel two-directed GLS and a simple diversification procedure. It will be shown that these strategies are capable of significantly limiting computation time and of increasing solution quality, making them useful for solving large routing problems in general.

2.1. Phase 1: constructing initial solutions

At the beginning of the search, a single initial solution is created by a modification of the savings heuristic (Clarke & Wright (1964)). As in the original savings heuristic, the search is started by serving each customer individually. There are three differences in comparison to the original savings heuristic. First, as in Liu and Shen (1999), the heuristic is implemented from an insertion point of view, i.e., when merging two routes R_1 and R_2, the search is not limited to inserting R_1 either before or after R_2, but in addition positions between consecutive customers within route R_2 are considered. Second, the calculated savings take into account both vehicle costs and total distance. Vehicle sizes are updated whenever needed and always set to the smallest vehicle available capable of serving the customers on the route. Third, the mergers are limited to the p closest routes only. The geographical proximity is based on the Euclidean distance of the average X and Y coordinates of the customers on the routes. Each time m route mergers have been executed, the information on the geographically close routes is updated. Moreover, after selecting two geographically close routes, R_1 and R_2, only the c customers from R_2 which are closest to the endpoints of R_1 are considered. The limit distance for the c^{th} closest customer is determined at the beginning of the search for all customers and maintained in memory. During the search only a comparison to the limit value is used to determine whether a given customer v is among the c closest. Here p, m and c are user-defined parameters. To save time, the calculated savings are stored in a matrix, which is updated during the search based on the mergers performed. Mergers are attempted until no further improvement can be found.

2.2. Phase 2: route elimination

The second phase focuses on minimizing the number of routes in the created initial solution. The applied procedure is called ELIM and is based on simple customer reinsertions. In ELIM, all routes of the current solution are considered for depletion in random order and eliminations are attempted until no more improvements can be found. For a given route, ELIM removes all customers in the order they are currently served, and tries to insert them in the p geographically closest neighboring routes, in the same way as in the initial solution heuristic. The geographical closeness of the routes is calculated both before and after Phase 2. For a given customer v and geographically close route R_2, only insertion positions adjacent to one of the c closest customers with regard to v are considered and the best feasible insertion according to the total cost objective is selected. If all the removed customers have been inserted in other routes at a lower cost, the route is eliminated; otherwise the executed insertions are backtracked.

2.3. Phase 3: local search improvement

2.3.1. Local searches

The solution from Phase 2 is further improved in Phase 3 by four local search heuristics that are guided by the TA and GLS metaheuristics. In addition to the above described ELIM procedure, the four local search operators include a route splitting operator called SPLIT, and ICROSS and IOPT operators suggested in Bräysy (2003). ICROSS and IOPT are extended here with the adjustment of vehicle types and costs.

The SPLIT neighborhood consists of all solutions that result from splitting a single route in the current solution into two parts at any point. We employ it in a 'greedy', first-accept fashion, simply by looping through all routes (in the order of the given fixed route numbers) and all customers in them, splitting the selected route into two parts at the position of the current customer. The move is made if the split reduces total cost. After a successful split, the information on the geographically close routes is updated. Here SPLIT is applied only every third iteration.

ICROSS is a generalization of CROSS exchanges (Taillard, Badeau, Gendreau, Guertin, & Potvin, 1997) and works by relocating or exchanging segments of consecutive customers from two different routes. The maximum segment length is limited to l_s customers. Compared to previous work, ICROSS is modified here so that only geographically close routes and only segments that involve the geographically closest customer pairs in the two routes are considered. To be more precise, the geographically close routes are defined in the same way as in Phases 1 and 2, but for a given route R_1, its p geographically closest routes are considered in random order to better diversify the search. For a given customer v, currently served by R_1, ICROSS first checks all customers w in R_2 that are among c closest from v and whose time window matches v. The time windows match if $E_v + S_v + \mathrm{TIME}(v,w) \leqslant L_w$ where E_v is the earliest possible service time of v, S_v is the service time of v, $\mathrm{TIME}(v,w)$ is the travel time between v and w, and L_w is the latest possible service time of w. Then, the resulting (v,w) pairs are sorted in ascending order according to their Euclidean distance, and ICROSS moves are attempted only for segments that start from v

or w and only for insertion positions adjacent to v or w and only for q closest (v,w) pairs, starting from the closest pair.

The IOPT intra-tour operator is a generalization of Or-opt (Or, 1976). It considers segments up to a given maximum length l_s and also includes moves where the segment is reversed before it is relocated (Bräysy, 2003). Here the search is limited so that only insertion positions adjacent to c closest customers with regard to the segment endpoints are considered. Moreover, in our implementation, IOPT is applied only every second iteration to further limit the search.

The four local search operators are employed with the first-accept strategy in the following order: ICROSS, IOPT, ELIM, and SPLIT and are repeated for a given number of iterations n_{improve}. Note that as in Bräysy et al. (2007), ELIM is applied only every second iteration. To increase the efficiency of the improvement phase, the local searches do not consider a route pair or a single route if no improvement was found last time and the routes have not changed since.

2.3.2. Metaheuristic frameworks

The above described local searches are embedded in the TA and GLS metaheuristic frameworks that are used simultaneously. The basic idea of TA is to allow also local search moves that worsen the objective value, as long as the worsening is within the current value of the threshold limit. The threshold limit T, is adjusted during the search. For more details, see Dueck and Scheurer (1990).

The algorithm starts with threshold $T = 0$ (no worsening allowed) and is repeated with that value until a local minimum is reached. Then, T is set to a new maximum value, $T = r \cdot T_{\max}$ where r is a random number in the range $[0,1]$ and $T_{\max}$ is a user-defined parameter. At each iteration, the T is reduced by ΔT units until $T = 0$. Here $\Delta T = r \cdot \Delta T_{\max}$ and $\Delta T_{\max}$ is a parameter. Note that, to diversify the search, ΔT is random as opposed to standard deterministic reduction. When $T = 0$, the search is repeated with zero threshold until no more improvements can be found. After that, T is set to $T = r \cdot T_{\max}$ again and so on until n_{improve} iterations are tried. If no more improvements have been found for a given number of iterations n_{restart}, the search is restarted from the current best solution and the threshold is set to $T = r \cdot T_{\max}$. The threshold is also set to zero each time a new best solution is found to intensify the search.

By further analyzing the algorithm proposed by Bräysy et al. (2007), we noticed that when the value of T was above zero, i.e., when worsening was allowed, repeating ICROSS operator for all route pairs resulted in significant overall worsening of solution quality and even a disruption of the structure of the current solution. Moreover, many local moves were required to restore the current solution back to a "good" level. Therefore, it was decided to limit the number of worsening ICROSS moves. To be more precise, when $T > 0$ only k routes are randomly selected as route R_1 and ICROSS is applied to p closest routes to R_1 in random order. Here the random order for the closest routes is critical to maintain the diversity of the search. As a result, a significantly smaller and more locally restricted worsening is allowed. Moreover, for SPLIT, the worsening is not allowed as it was discovered that it resulted in too a large diversification.

On the other hand, it was noted that the new locally restricted search strategy ended up intensifying the search too strongly and lacked systematic diversification. To deal with this issue, we included a new two-directed GLS metaheuristic.

GLS operates by augmenting the objective function with a penalty term based on particular solution features (long edges) not considered to be part of a near-optimal solution (see Voudouris & Tsang (1998) for more details). In our case the GLS is used to guide the local search procedures by defining a modified distance matrix that is used to evaluate the moves. In the beginning the modified distance matrix equals the original Euclidean distance matrix. Each time when $T = 0$ and no more improvements have been found and with each restart from the current best solution, two arc distances are adjusted and stored in the modified distance matrix. As is in the standard GLS, the first arc is penalized by increasing its length. In our implementation, however, the second arc is favored by reducing its length. More precisely, the penalized arc is selected according to maximum utility function

$$U = \frac{\text{DIST}(i,j)}{(1+p_{ij})} \tag{1}$$

and the favored arc is defined as the shortest arc that has not been selected before. The distances of the chosen arcs are modified as

$$\text{DIST}_P^{\text{New}}(i,j) = \text{DIST}(i,j) \cdot (1+\lambda) \tag{2}$$

$$\text{DIST}_F^{\text{New}}(i,j) = \text{DIST}(i,j) \cdot (1-\lambda) \tag{3}$$

where λ is a parameter value whose value is reset to $\lambda = r \cdot \lambda_{\max}$ at each restart from the current best solution. Moreover, with the restart, the modified distance matrix is initialized to original values with 70% probability. Finally, to further diversify the search efforts, we reset the l_s to a larger value every dth iterations.

We used here the same general search limitation strategies as in Bräysy et al. (2007), i.e., the double-linked list in an array data structure (Kytöjoki, Nuortio, Bräysy, & Gendreau, 2007), standard time window feasibility check techniques (Campbell & Savelsberg, 2004), and the opportunistic constraint feasibility check strategy. The latter means that features that are easiest or quickest to calculate or that are most likely to violate a feasibility condition are checked first. For more details, we refer to Bräysy et al. (2007).

3. Computational results

Computational experiments were performed to examine the performance of the proposed algorithm. We first describe the instances considered in these experiments, as well as the parameter values used. This is followed by a presentation and a discussion of the results obtained with our metaheuristic.

3.1. Experimental setting and parameter values

The computational experiments were performed using the benchmark instances proposed by Liu and Shen (1999) and 600 new benchmark instances suggested in this paper. In contrast to Liu and Shen, the sum of all vehicle costs and total distance is considered as the optimization objective, as opposed to the sum of vehicle costs and *en route* time. The new objective was first introduced in Bräysy et al. (2007) and it is believed to be of a higher practical value than the former objective function. The Liu and Shen benchmarks are derived from the well-known VRPTW instances of Solomon (1987). Solomon's problem sets for the VRPTW consist of 56 instances of 100 customers with randomly generated coordinates (set R), clustered coordinates (set C) or both (semi-clustered RC set). The difference between Subsets R1, C1 and RC1 and R2, C2 and RC2 lie in the vehicle capacities and scheduling horizon. For each six subsets Liu and Shen introduced several vehicle types with different capacities and costs. In addition, three different vehicle cost structures *A*, *B* and *C* were suggested so that cost structure *A* refers to the largest vehicle costs and *C* to the smallest. To limit the computational tests, we omitted here the cost structure B, resulting in 112 test problems of 100 customers each.

The suggested new test problems are based on the large-scale VRPTW benchmark instances of Gehring and Homberger (1999). Similar to Solomon, Gehring and Homberger constructed (1987) random, clustered and semi-clustered problem sets, consisting of 200, 400, 600, 800 and 1000 customers, so that there are 60

Table 1
Vehicle costs and capacities used for each problem set.

Capacity	Cost	Capacity	Cost	Capacity	Cost
C1		*C2*		*R1*	
40	200	120	575	40	140
70	335	240	1100	70	230
100	460	350	1540	100	310
140	615	470	1975	140	405
170	715	580	2320	170	460
200	800	700	2700	200	500
240	910	820	2955	240	550
270	975	930	3160	270	565
R2		*RC1*		*RC2*	
170	590	40	125	170	590
340	1115	70	205	340	1115
500	1550	100	275	500	1550
670	1945	140	355	670	1945
840	2270	170	420	840	2270
1000	2500	200	450	1000	2500
1170	2690	240	495	1170	2690
1330	2795	270	500	1330	2795

problems of each size and 10 problems in each of the above groups. In total there are 300 problems. For the Gehring and Homberger instances, we suggest a set of vehicle types and costs. The same 8 vehicle types and costs are used for every problem size. The vehicle capacities and costs differ only between the six problem sets, as detailed in Table 1. The vehicle capacities and costs in Table 1 were defined as follows. As in Liu and Shen (1999), we have used the maximum capacity in the corresponding VRPTW instance, V_B as the starting point. The cost of the vehicle with a carrying capacity V_B is the same as for the corresponding Liu and Shen (1999) 100-customer problem set. The other vehicle capacities and costs are based on real-life information collected from Finland. More precisely, we first surveyed the most typical truck types available and fixed costs related to them. Excluding vans, we found out that there are eight common vehicle types available. Liu and Shen (1999) defined only vehicle capacities smaller than the original VRPTW maximum capacity, making the problems somewhat easier to solve (optimizing the capacity utilization of larger vehicles is often harder). Here we decided also to enable vehicles with larger capacity than in the original VRPTW problem, so that we set the sixth largest vehicle carrying capacity (9 tons) to equal the V_B. As a result, there are two truck types larger than V_B and five that are smaller in each problem. The other capacities were defined using direct relation wrt. V_B so that e.g. the capacity equaling 12 tons is obtained by multiplying V_B by 12/9 and rounding up or down with accuracy of 5 units. The costs of other vehicle types apart from V_B were defined by first analyzing the cost relations wrt. the 9 ton vehicle. We noticed in the analysis of the real-life capacity and cost data that apart from the 2–3 smallest vehicle types, there are linear economies of scale in the cost per capacity unit. The real-life costs of the smallest vehicle types were clearly relatively more expensive than the larger ones. Therefore, they were hardly ever used in the tests done with the preliminary data. Based on this information, we decided to apply the same linear economies of scale structure over all 8 vehicle types to improve the quality of the benchmarks, by using the scaling factor defined with the three largest vehicle types. The obtained costs were then rounded to the nearest 10. This corresponds to cost structure A. As in Liu and Shen (1999), we obtained cost structure C by dividing the vehicle costs by 10, resulting in 600 test instances. Here the relationship between fixed costs and vehicle capacity is more realistic than the cost structure by Liu and Shen (1999), where fixed costs in some cases are obviously too small in comparison to vehicle capacity.

We tested the performance of the algorithm with two different parameter settings that we denote here as *Quick* and *Normal*. In the *Normal* setting the above described parameter values were set as follows: $p = 10$, $c = 55$, $q = 25$, $n_{\text{improve}} = 4000$, $n_{\text{restart}} = 40$, $T_{\max} = 0.06$, $\Delta T_{\max} = 0.009$, $l_s = 3$, $\lambda_{\max} = 0.03$, $k = 3$, $m = 30$ and after each $d = 65$ iterations, l_s is set to $l_s = 5$. Moreover, the GLS is not applied during the last 1000 iterations. For the *Quick* setting only three parameter values are different: $p = 5$, $q = 15$ and $n_{\text{improve}} = 1000$. The computational testing was executed on Intel Core Duo T7700 (2.4 GHz) laptop computer with 2 GB memory.

3.2. Sensitivity analysis

In this section, we analyze the performance and sensitivity of the algorithm w.r.t. different parameter values. All results in this section are based on the average output over all 100-customer benchmarks with cost structure A and *Normal* setting.

The formatting of Figs. 1–5 is the same. The CPU time is plotted by a dotted line on the leftmost vertical axis, whereas the solution

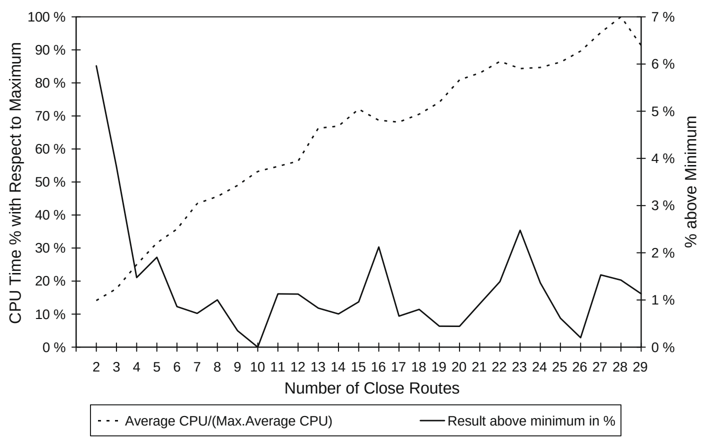

Fig. 1. The effect of close routes (p) with respect to relative CPU time and total cost excess in %.

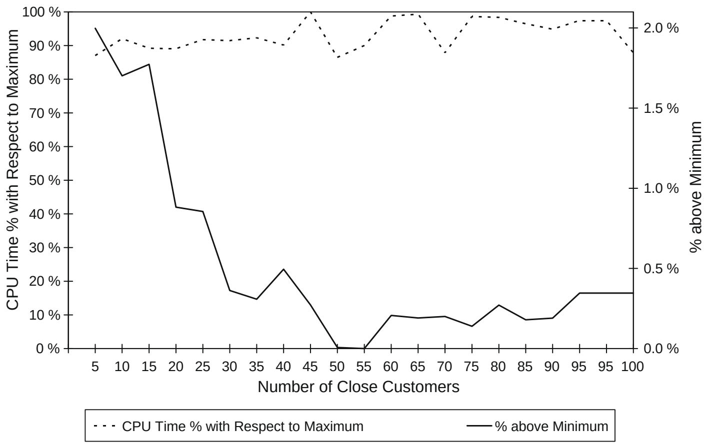

Fig. 2. The impact of number of close customers (c) to relative CPU time and total cost excess in %.

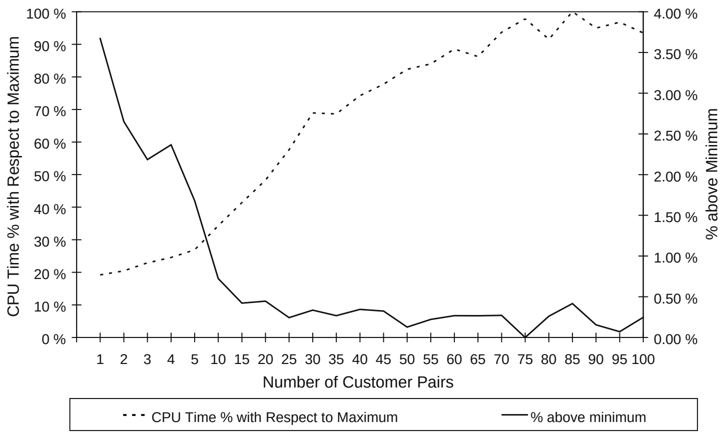

Fig. 3. The sensitivity of relative CPU time and cost excess wrt. parameter q.

quality is shown by the continuous line plotted on the rightmost vertical axis, for the different parameter values listed in on the horizontal axis. Both CPU time and solution quality are measured against respectively the largest CPU time and the lowest total cost.

In Fig. 1 we illustrate the impact of the number of geographically close routes p considered in the local searches. Based on the figure, it appears that at least 5 routes should be considered and that 10 is the best value. In some cases increasing the value above 10 may worsen the objective function value by as much as 2%. This is probably due to a too large diversification, caused by accepting more worsening moves. As can be expected, the CPU time grows steadily when increasing the value of p.

Fig. 2 depicts the effect of parameter c, i.e., the number of close customers considered in the search. As can be seen from the Figure, values 50-55 appear to be the best. Too small values (less than 30) clearly reduce the solution quality, but increasing the value over 55 does not affect the solution quality much. Surprisingly, the parameter values do not have much effect on the CPU time, probably because of other, more restrictive search limitations.

In Fig. 3, we illustrate the sensitivity of the output w.r.t. the number of closest customer pairs q, considered by the ICROSS local search. As one can expect, the higher the value of q, the longer the CPU time and the higher the solution quality. However, it appears that the solution quality does not improve much beyond $q = 25$. It is also interesting to note that sometimes a higher value can result in a slightly worse output.

Fig. 4 shows the impact of the maximum threshold value, $T_{\max}$, specifying an upper boundary for the actually used threshold values. It seems that values in range 0.04–0.06 are the best and values smaller and higher than this result in clearly worse output. It is also interesting and surprising to note that the CPU time becomes smaller with the higher $T_{\max}$ values even though this results in accepting more worsening moves. This is probably caused by the limitation whereby the algorithm ignores routes and route pairs

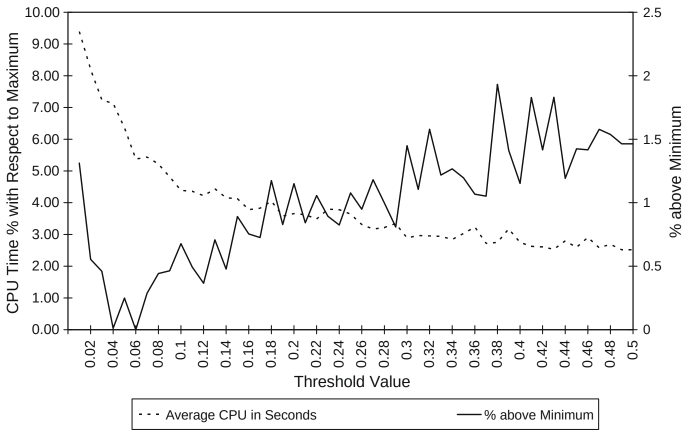

Fig. 4. The impact of the T_{max}value on the relative CPU time and total cost excess.

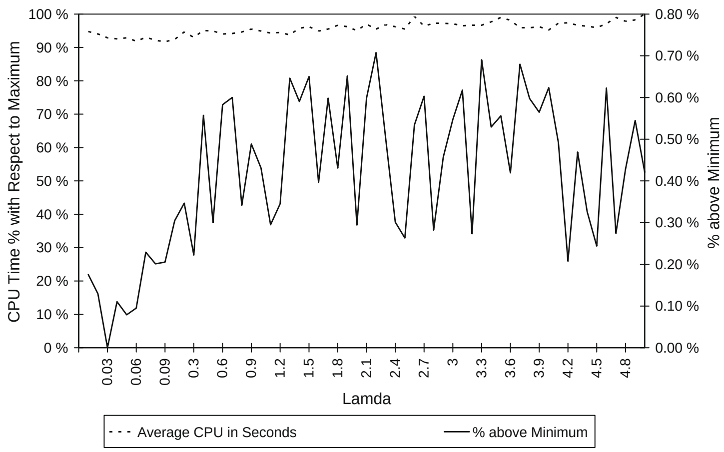

Fig. 5. The effect of parameter $\lambda$to relative CPU time and total cost excess.

that have not improved in the last iteration. As a result, the improvement process ends prematurely.

In Fig. 5 we consider the impact of parameter λ on the results. The parameter λ affects mainly how much the distances of the penalized and favoured arcs are modified. It appears that the differences between the different values are very small and 0.03 is the best value. Based on the figure, it can also be concluded that λhas little impact on CPU time.

In Fig. 6 we illustrate the convergence of the algorithm as a function of the iterations. As expected, the largest gains are obtained at the beginning. After the first 1000 iterations, on the average about 1% improvement appears possible. On the other hand, the fact that improvements are obtained up to 19,000 iterations also illustrates the power of the algorithm. It can also be said that the quality of the initial solutions is probably rather good as with 19,000 iterations only an 1.8% improvement proves possible.

Table 2 shows the impact of the applied TA and GLS metaheuristics. The values in the table correspond to the excess of total cost w.r.t. the best solutions found, over all 100-customer problems. We list in the table the minimum, average and worst output over 5 test runs for 4 different settings. TA and GLS means that both TA and GLS are applied, column TA refers to using only TA without GLS and column GLS using only GLS. Finally the rightmost column refers to ignoring both TA and GLS. It appears that the best results are obtained when both TA and GLS are applied. The difference in the best results compared to using TA is however just 0.17%. The best output of GLS appears to be 0.61% above the best value. Surprisingly, when neither TA nor GLS are used, the results are only 1.14% above the best output. The differences in the average and worst outputs appear, however, more significant, indicating the importance of the metaheuristic frameworks.

In addition to the different combinations of metaheuristics, we also tested the impact of resetting the threshold to $T = 0$ immedi-

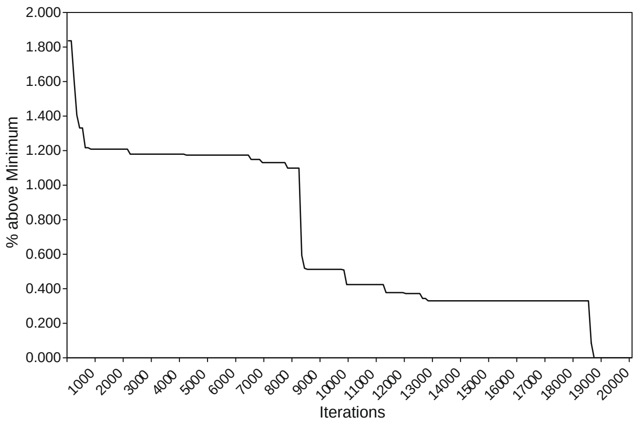

Fig. 6. Convergence of the total cost wrt. parameter n_{improve}.

Table 2
The impact of TA and GLS metaheuristics on the results.

	% Above minimum			
	TA & GLS	TA	GLS	Local search
Minimum	0.00	0.17	0.61	1.14
Average	0.87	1.53	1.99	3.76
Maximum	3.15	4.50	5.43	6.99

ately a new best-known solution is found, and noticed that it improved the total cost 0.6% on the average.

In Fig. 7 we describe the scaling of the suggested *Normal* and *Quick* variants over the tested problem sizes. It appears that both variants scale very well and are therefore well-suited to solving large-scale problems. The differences between the cost structures also appear very small. For *Quick* setting we present here the average of the cost structures A and C as they were almost identical and hard to plot separately.

In Table 3 we compare the best, worst and average outputs and CPU times of the *Normal* and *Quick* variants over the different problem sizes and cost structures. In general, it appears that the variance of the results over the five test runs is quite moderate, for both settings, given the significant number of random components in the algorithm.

3.3. Comparison of results

In this section we compare and analyze the actual results of the *Quick* and *Normal* settings to the tested 712 benchmark problems, and compare the results with the previous method of Bräysy et al. (2007). The method of Bräysy et al. (2007) reported 167 best-known solutions to the 168 benchmarks of Liu and Shen (1999), thus giving a good comparison point for the performance. Here we resolved all the problems with the algorithm of Bräysy et al. (2007), denoted here by MSDA using the same computer as for the new *Normal* and *Quick* settings to enable direct comparison. As the CPU times of MSDA are considerably higher, we also created a new limited variant of the method of Bräysy et al. (2007), denoted by MSDAL where the local search algorithms are limited to consider only customers among 50 closest or less, depending on

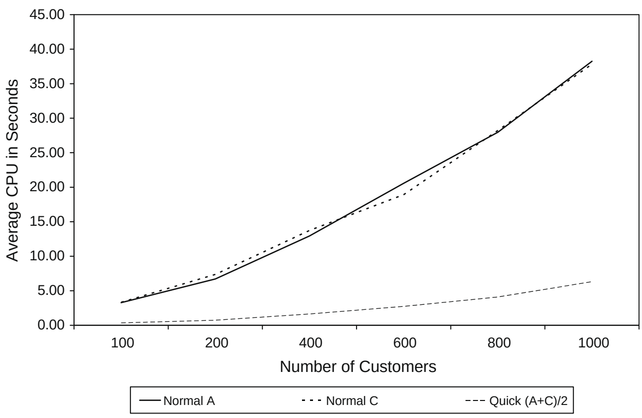

Fig. 7. The scaling of the CPU time wrt. problem size.

Table 3
The best, worst and average output of the normal and quick variants.

Size	Cost	Normal				Quick			
		Best	Average	Worst	CPU	Best	Average	Worst	CPU
100	A	4756.08	4773.39	4788.56	3.27	4771.83	4800.91	4827.81	0.34
100	C	1430.76	1448.35	1460.02	3.32	1447.91	1464.41	1477.23	0.35
200	A	13588.70	13851.36	14049.24	6.71	13923.96	14214.91	14431.91	0.75
200	C	3829.72	3866.98	3896.91	7.36	3888.61	3952.11	3994.44	0.73
400	A	28337.82	28873.42	29309.20	12.94	29195.41	29736.83	30127.16	1.66
400	C	8502.17	8582.99	8644.00	13.73	8751.84	8894.66	8997.85	1.63
600	A	46498.11	47283.38	47824.59	20.56	48197.18	49044.98	49630.11	2.76
600	C	16036.31	16188.07	16297.23	18.91	16748.80	17036.71	17243.14	2.71
800	A	66312.86	66970.33	67492.85	27.97	69182.82	70255.45	70962.00	4.15
800	C	25994.01	26183.24	26335.71	28.22	27518.43	27932.01	28202.94	4.05
1000	A	89165.95	90175.14	90935.15	38.25	93994.43	95174.56	96057.13	6.35
1000	C	38552.68	38860.24	39128.94	37.82	41250.37	41893.74	42362.48	6.29

Table 4
Results for the 100-customer problems.

Data set	Size	Cost	Normal	Quick	MSDAL	MSDA	Normal quick (%)	Normal MSDA (%)	Normal MSDA (%)	Quick-MSDA (%)	Quick-MSDA (%)	MSDA–MSDA (%)
C1	100	A	7085.91	7090.23	7087.20	7141.15	−0.06	−0.02	−0.77	0.04	−0.71	0.76
C2	100	A	5689.40	5688.60	5719.98	5797.38	0.01	−0.53	−1.86	−0.55	−1.88	1.35
R1	100	A	4060.96	4080.65	4074.73	4131.31	−0.48	−0.34	−1.70	0.15	−1.23	1.39
R2	100	A	3180.58	3205.98	3194.50	3310.70	−0.79	−0.44	−3.93	0.36	−3.16	3.64
RC1	100	A	4935.52	4975.33	4958.93	4948.53	−0.80	−0.47	−0.26	0.33	0.54	−0.21
RC2	100	A	4231.25	4233.13	4241.72	4399.12	−0.04	−0.25	−3.82	−0.20	−3.77	3.71
C1	100	C	1615.40	1617.97	1616.99	1622.03	−0.16	−0.10	−0.41	0.06	−0.25	0.31
C2	100	C	1185.69	1187.23	1186.33	1223.86	−0.13	−0.05	−3.12	0.08	−2.99	3.16
R1	100	C	1539.90	1559.07	1538.90	1579.17	−1.23	0.06	−2.49	1.31	−1.27	2.62
R2	100	C	1149.06	1168.47	1158.71	1257.65	−1.66	−0.83	−8.63	0.84	−7.09	8.54
RC1	100	C	1749.66	1790.99	1749.37	1758.29	−2.31	0.02	−0.49	2.38	1.86	0.51
RC2	100	C	1372.82	1391.67	1381.71	1566.01	−1.35	−0.64	−12.34	0.72	−11.13	13.34
Average			3149.68	3165.78	3159.09	3227.93	−0.75	−0.30	−3.32	0.46	−2.59	3.26
% above minimum			0.01	0.77	0.31	3.59						
Runs			5	5	3	3						
Average CPU seconds per instance			3.30	0.35	24.87	50.03						

the dynamically adjusted parameter value of MSDA that limits the search to only closest customers that have been part of improving moves (for more details, see Bräysy et al., 2007). For both MSDA and MSDAL 1000 iterations 3 runs were executed.

The results to the 100-customer instances are presented in Table 4. The table is divided in four parts. On the left the total cost for the *Normal* and *Quick* variants and for MSDA and MSDAL for the six problem groups are described. Here the described values are the best over 5 and 3 runs, respectively. The top part of the table gives results related to cost structure A, whereas the results for cost structure C are given in the lower part. On the right the perceptual differences between the methods pair wise are compared. Here a negative value means that the method listed first is better and a positive indicates the opposite. At the bottom we list the number of runs and the CPU time in seconds for a single run.

Based on the table, the new *Normal* and *Quick* methods are considerably (10–100 times) faster than the previous MSDA and MSDAL methods. It can also be concluded that *Quick* variant is about 10 times faster than *Normal*. At the same time *Normal* is consistently better in each problem group wrt. MSDA and MSDAL. Even the *Quick* variant is better than MSDA in each case. Surprisingly, the MSDAL reports significantly better solutions than MSDA of Bräysy et al. (2007). In sum there appears to be a clear benefit in limiting the solution space: *Quick*finds better solutions in a shorter period of computation time. This is also confirmed by the fact that *Quick* appears slightly better than *Normal* for group C2 and cost structure A.

The results for the 600 new benchmarks are summarized in Table 5. The table is divided by horizontal lines according to the problems sizes and used cost structures. Within each problem size and cost structure the results are averaged over the six problem groups and total cost value is reported. The comparison with the previous method of Bräysy et al. (2007) is done using the MSDAL variant because of its better performance and lower computational effort. The CPU times in seconds per problem group for each method are reported in the middle of the table. The detailed results to each individual problem are given in Appendixes I and II, with new best-known solutions marked in bold.

Based on Table 5, the *Normal* variant appears clearly the best, and on average about 4% better than the *Quick* variant and almost 2% better than the MSDAL. The differences also seem to increase with the problem size. From the computational viewpoint, the relative difference, *Quick* appears to be on average about 7 times faster than *Normal* and, as illustrated above, both new variants appear to scale well. However, the scaling of MSDAL appears a lot worse. With the 1000-customer problems it is as much as about 50 times slower than *Normal*.

4. Conclusions

In this paper, we have presented a new hybrid threshold-accepting and guided local search metaheuristic that is specifically designed for solving large-scale fleet size and mix routing prob-

Table 5
The results for the new extended benchmark problems.

Data set	Size	Cost	Normal	Quick	MSDAL	Normal-quick (%)	Normal-MSDAL (%)	Quick-MSDAL (%)
C1	200	A	17129.92	17361.03	17173.12	−1.33	−0.25	1.09
C2	200	A	17385.39	17495.07	17433.42	−0.63	−0.28	0.35
R1	200	A	11471.04	11748.31	11593.77	−2.36	−1.06	1.33
R2	200	A	12611.59	13307.42	12149.44	−5.23	3.80	9.53
RC1	200	A	10004.18	10212.95	10143.88	−2.04	−1.38	0.68
RC2	200	A	12930.10	13419.01	12475.09	−3.64	3.65	7.57
C1	200	C	4136.96	4161.14	4143.60	−0.58	−0.16	0.42
C2	200	C	3444.10	3510.08	3450.99	−1.88	−0.20	1.71
R1	200	C	4311.74	4365.76	4349.61	−1.24	−0.87	0.37
R2	200	C	3869.59	3921.21	3964.24	−1.32	−2.39	−1.09
RC1	200	C	3692.49	3747.80	3724.67	−1.48	−0.86	0.62
RC2	200	C	3523.42	3625.64	3590.36	−2.82	−1.86	0.98
C1	400	A	36490.01	36994.17	36543.15	−1.36	−0.15	1.23
C2	400	A	35737.00	35988.11	35943.01	−0.70	−0.57	0.13
R1	400	A	23804.33	24553.23	24033.19	−3.05	−0.95	2.16
R2	400	A	26468.35	28271.49	25013.81	−6.38	5.81	13.02
RC1	400	A	21278.46	21995.29	21489.04	−3.26	−0.98	2.36
RC2	400	A	26248.74	27370.15	25704.74	−4.10	2.12	6.48
C1	400	C	9980.16	10107.31	9937.46	−1.26	0.43	1.71
C2	400	C	7236.69	7408.02	7373.82	−2.31	−1.86	0.46
R1	400	C	9469.58	9765.34	9553.84	−3.03	−0.88	2.21
R2	400	C	8288.01	8589.69	8651.97	−3.51	−4.21	−0.72
RC1	400	C	8612.17	8898.96	8674.91	−3.22	−0.72	2.58
RC2	400	C	7426.39	7741.69	7756.08	−4.07	−4.25	−0.19
C1	600	A	58717.06	59726.60	58917.90	−1.69	−0.34	1.37
C2	600	A	56406.58	56868.07	56969.27	−0.81	−0.99	−0.18
R1	600	A	40400.51	42462.11	41167.05	−4.86	−1.86	3.15
R2	600	A	44832.54	47179.45	43512.70	−4.97	3.03	8.43
RC1	600	A	35583.46	37173.45	36205.54	−4.28	−1.72	2.67
RC2	600	A	43048.52	45773.39	43120.29	−5.95	−0.17	6.15
C1	600	C	18322.81	18661.65	18263.58	−1.82	0.32	2.18
C2	600	C	12910.40	13328.83	13129.29	−3.14	−1.67	1.52
R1	600	C	18951.98	19938.79	19329.31	−4.95	−1.95	3.15
R2	600	C	15466.30	16282.71	17116.21	−5.01	−9.64	−4.87
RC1	600	C	16534.05	17393.78	16878.11	−4.94	−2.04	3.06
RC2	600	C	14032.34	14887.04	14840.67	−5.74	−5.45	0.31
C1	800	A	83616.82	85376.59	84031.98	−2.06	−0.49	1.60
C2	800	A	76440.60	76906.34	78659.84	−0.61	−2.82	−2.23
R1	800	A	60417.24	64141.29	61776.80	−5.81	−2.20	3.83
R2	800	A	64267.07	67610.64	63441.08	−4.95	1.30	6.57
RC1	800	A	53505.16	56458.45	54658.17	−5.23	−2.11	3.29
RC2	800	A	59630.30	64603.59	61382.78	−7.70	−2.86	5.25
C1	800	C	30516.92	31182.73	30628.48	−2.14	−0.36	1.81
C2	800	C	18945.96	19642.02	19359.69	−3.54	−2.14	1.46
R1	800	C	31845.44	34141.85	32541.16	−6.73	−2.14	4.92
R2	800	C	25010.18	26567.91	27241.67	−5.86	−8.19	−2.47
RC1	800	C	28122.31	30080.51	28856.23	−6.51	−2.54	4.24
RC2	800	C	21523.29	23495.57	23015.30	−8.39	−6.48	2.09
C1	1000	A	115023.57	117816.15	116020.70	−2.37	−0.86	1.55
C2	1000	A	98576.46	99600.16	102138.68	−1.03	−3.49	−2.49
R1	1000	A	82919.62	88660.99	85609.66	−6.48	−3.14	3.56
R2	1000	A	86196.78	92323.20	89037.40	−6.64	−3.19	3.69
RC1	1000	A	73488.49	78563.27	76032.60	−6.46	−3.35	3.33
RC2	1000	A	78790.75	87002.82	82336.65	−9.44	−4.31	5.67
C1	1000	C	48902.05	49809.27	48866.57	−1.82	0.07	1.93
C2	1000	C	26442.64	27222.84	27378.82	−2.87	−3.42	−0.57
R1	1000	C	46936.79	50599.96	48358.16	−7.24	−2.94	4.64
R2	1000	C	35683.24	38901.54	40139.08	−8.27	−11.10	−3.08
RC1	1000	C	42260.22	45919.04	44174.03	−7.97	−4.33	3.95
RC2	1000	C	31091.16	35049.55	34425.39	−11.29	−9.69	1.81
Average			33681.83	35265.18	34407.13	−4.01	−1.86	2.27
% above minimum			0.34	4.60	2.32			
Runs			5	5	5			
Average CPU seconds per instance			21.25	3.11	839.1			

lems with time windows. The central part of the described algorithm consists of different strategies for balancing a limitation and intensification of the search. The computational tests were done with the benchmarks of Liu and Shen (1999) and on 600 new large-scale real-life based benchmarks suggested in this paper. A comprehensive computational study, including detailed sensitivity analysis showed that the suggested method is competitive with the previous best approach and scales almost linearly for problems up to 1000 customers.

Acknowledgement

This work was partially supported by TEKES and the EDGE project funded by the Research Council of Norway. This support is gratefully acknowledged.

Appendix I

				NORMAL		QUICK		MSDAL			NORMAL		QUICK		MSDAL	
Type	Size	No.	Cost	Best	CPU	Best	CPU	Best	CPU	Cost	Best	CPU	Best	CPU	Best	CPU
C1	100	1	A	7097.93	3.09	7104.69	0.24	**7096.13**	26.28	C	**1628.94**	3.01	**1628.94**	0.28	**1628.94**	32.30
C1	100	2	A	**7085.47**	3.66	7089.29	0.29	7086.19	26.67	C	**1597.66**	2.95	1597.66	0.28	1597.66	33.73
C1	100	3	A	**7080.41**	3.67	7084.93	0.29	7081.89	25.49	C	**1596.56**	3.06	**1596.56**	0.29	1605.35	39.43
C1	100	4	A	**7075.06**	3.45	7075.96	0.29	7075.89	26.57	C	**1590.86**	3.09	**1590.86**	0.29	1594.14	27.96
C1	100	5	A	7096.22	3.15	7111.36	0.27	**7095.13**	30.31	C	**1628.94**	2.97	**1628.94**	0.28	**1628.94**	33.19
C1	100	6	A	7088.35	3.63	7092.04	0.29	**7087.58**	31.35	C	**1628.94**	3.01	1628.94	0.28	**1628.94**	34.22
C1	100	7	A	**7090.91**	3.17	7091.39	0.25	7092.83	28.21	C	**1628.94**	2.96	**1628.94**	0.29	**1628.94**	32.74
C1	100	8	A	**7081.18**	3.72	7081.83	0.29	7087.12	28.57	C	**1622.75**	3.13	1631.98	0.28	1624.07	40.52
C1	100	9	A	**7077.68**	3.94	7080.59	0.31	7082.02	23.33	C	**1614.99**	3.28	1628.94	0.29	1615.93	50.10
C2	100	1	A	5700.87	2.61	5696.86	0.47	5695.02	13.31	C	**1194.33**	4.10	**1194.33**	0.55	**1194.33**	24.60
C2	100	2	A	5689.70	2.70	**5687.07**	0.45	5689.81	11.40	C	**1185.24**	3.10	1187.07	0.48	1189.35	16.11
C2	100	3	A	**5681.55**	2.47	**5681.55**	0.42	5697.21	13.83	C	**1176.25**	3.33	1177.80	0.57	1176.27	17.77
C2	100	4	A	5677.69	2.18	**5677.66**	0.40	5693.93	18.20	C	1176.55	3.51	1177.66	0.50	1177.80	19.26
C2	100	5	A	**5691.70**	2.63	**5691.70**	0.46	5726.37	7.99	C	**1190.36**	3.48	1191.70	0.44	1191.58	15.52
C2	100	6	A	**5691.70**	2.32	**5691.70**	0.46	**5691.70**	8.84	C	1188.62	3.52	1190.36	0.50	**1188.62**	16.49
C2	100	7	A	**5692.36**	2.21	5692.70	0.44	5694.64	10.63	C	1187.71	2.80	1189.35	0.51	**1184.88**	15.75
C2	100	8	A	**5689.59**	2.93	**5689.59**	0.43	5871.14	9.79	C	**1186.50**	2.69	1189.59	0.46	1187.86	12.88
R1	100	1	A	**4342.72**	3.36	4356.95	0.27	4355.46	33.43	C	1951.89	3.58	1967.17	0.27	**1951.22**	35.66
R1	100	2	A	4189.21	4.17	4217.28	0.33	**4182.47**	38.51	C	**1778.29**	3.49	1790.52	0.28	1782.75	38.30
R1	100	3	A	**4051.62**	3.81	4060.58	0.32	4060.33	38.97	C	**1555.26**	3.45	1563.11	0.29	1555.97	39.52
R1	100	4	A	**3972.65**	4.01	3985.27	0.31	3990.04	32.08	C	1372.08	3.44	1389.18	0.30	**1361.46**	35.98
R1	100	5	A	**4152.50**	3.64	4200.93	0.30	4169.28	36.15	C	1698.26	3.59	1723.95	0.30	**1698.25**	34.92
R1	100	6	A	4085.30	3.70	4099.41	0.32	**4085.07**	35.52	C	**1590.11**	3.49	1616.92	0.27	1598.48	36.70
R1	100	7	A	**3996.74**	3.94	4011.44	0.35	4018.07	37.40	C	1439.81	3.59	1459.53	0.29	**1435.41**	33.00
R1	100	8	A	**3949.50**	3.78	3977.11	0.30	3973.46	36.12	C	**1334.68**	3.30	1341.49	0.30	1336.69	38.71
R1	100	9	A	**4035.89**	3.84	4064.76	0.33	4053.79	34.38	C	1514.13	3.63	1547.20	0.31	**1507.10**	35.28
R1	100	10	A	**3991.63**	3.81	3997.90	0.34	4023.15	36.18	C	1461.85	3.62	1481.93	0.28	1465.83	34.65
R1	100	11	A	4009.61	3.89	4029.78	0.33	**4008.88**	36.95	C	1439.14	3.63	1462.97	0.29	**1435.93**	36.45
R1	100	12	A	3954.19	3.87	3966.44	0.33	3976.79	37.79	C	1343.26	3.51	1364.84	0.30	1337.68	34.05
R2	100	1	A	3530.24	3.32	3617.20	0.45	**3529.00**	16.42	C	1466.13	3.29	1508.62	0.36	**1439.76**	20.13
R2	100	2	A	**3335.61**	3.16	3384.49	0.41	3351.30	15.67	C	1296.78	3.26	1340.00	0.44	**1288.70**	18.50
R2	100	3	A	3164.03	2.32	3178.88	0.43	**3162.84**	15.38	C	**1127.28**	3.45	1136.68	0.44	1170.93	22.82
R2	100	4	A	**3029.83**	2.79	3040.11	0.39	3063.01	26.71	C	**1000.89**	3.87	1008.85	0.46	1030.24	30.54
R2	100	5	A	3261.19	4.12	3286.44	0.47	**3252.43**	12.28	C	1240.74	3.06	1272.12	0.46	**1232.62**	17.96
R2	100	6	A	**3165.85**	3.01	3189.59	0.45	3181.43	15.44	C	**1141.13**	2.75	1177.59	0.38	1161.23	15.65
R2	100	7	A	3102.79	2.30	**3100.64**	0.42	3114.25	23.46	C	**1067.97**	2.84	1073.22	0.46	1082.16	22.37
R2	100	8	A	**3009.13**	2.56	3019.44	0.40	3062.73	12.62	C	**979.50**	3.21	985.81	0.44	994.23	17.07
R2	100	9	A	3155.60	3.03	3162.99	0.45	**3141.17**	13.49	C	1140.38	3.41	1140.31	0.45	**1123.19**	19.55
R2	100	10	A	**3206.09**	2.40	3216.11	0.42	3214.19	12.59	C	**1170.29**	2.78	1184.58	0.45	1186.73	16.11
R2	100	11	A	**3026.02**	3.31	3069.96	0.41	3067.17	13.97	C	**1008.54**	3.28	1025.40	0.47	1035.99	16.20
RC1	100	1	A	**5168.23**	2.99	5255.67	0.26	5236.09	27.86	C	2053.55	3.56	2090.05	0.26	**2050.37**	29.88
RC1	100	2	A	**5025.22**	3.59	5075.61	0.26	5047.26	31.95	C	**1872.49**	3.56	1898.37	0.26	1877.63	29.87
RC1	100	3	A	4888.53	3.56	4921.64	0.30	4894.11	29.99	C	1663.08	3.27	1731.48	0.26	1662.70	29.91
RC1	100	4	A	**4747.38**	3.49	4775.64	0.28	4756.38	31.05	C	1540.61	3.35	1571.26	0.27	1543.47	29.51
RC1	100	5	A	5068.54	3.17	5094.70	0.26	**5065.31**	30.09	C	**1929.89**	3.78	1981.18	0.27	1930.84	32.01
RC1	100	6	A	4972.11	3.41	5021.54	0.28	4999.75	31.45	C	1776.52	3.34	1827.42	0.24	**1761.63**	26.20
RC1	100	7	A	4861.04	3.53	4878.50	0.29	4871.35	27.42	C	1633.29	3.39	1678.88	0.25	1624.91	27.88
RC1	100	8	A	4753.12	3.45	4779.33	0.30	4801.24	27.55	C	**1527.87**	3.32	1549.28	0.28	1543.38	26.37
RC2	100	1	A	4404.07	3.41	**4398.21**	0.28	4417.23	14.09	C	**1630.53**	3.42	1672.36	0.33	1637.98	22.12
RC2	100	2	A	**4266.96**	3.30	4270.13	0.28	4281.78	13.98	C	1461.44	3.23	1466.68	0.30	**1460.74**	19.50
RC2	100	3	A	4189.94	3.24	**4185.70**	0.29	4196.53	13.91	C	**1292.92**	2.96	1296.19	0.41	1297.74	21.00
RC2	100	4	A	**4098.34**	3.22	4109.55	0.29	4114.17	16.13	C	**1162.91**	2.98	1167.67	0.45	1172.86	15.41
RC2	100	5	A	**4304.52**	3.41	4308.33	0.28	4305.65	12.70	C	**1532.67**	3.00	1548.52	0.33	1538.95	18.33
RC2	100	6	A	**4272.82**	3.68	4296.90	0.28	4285.31	13.81	C	**1420.89**	3.98	1463.99	0.34	1428.22	16.57
RC2	100	7	A	4219.52	3.19	**4213.66**	0.29	4217.81	12.61	C	**1328.29**	4.11	1353.19	0.36	1351.32	16.75
RC2	100	8	A	4093.83	3.00	**4082.58**	0.27	4115.24	10.38	C	1152.92	3.27	1164.79	0.39	1165.90	14.57

Appendix II

				Normal		Quick		MSDAL			Normal		Quick		MSDAL	
Type	Size	No.	Cost	Best	CPU	Best	CPU	Best	CPU	Cost	Best	CPU	Best	CPU	Best	CPU
C1	200	1	A	**17456.15**	7.55	17597.57	0.68	17478.35	95.30	C	**4216.08**	8.34	4221.67	0.74	4216.08	106.09
C1	200	2	A	17095.62	7.58	17267.08	0.78	**17056.41**	96.25	C	**4164.70**	7.58	4182.80	0.77	4165.48	119.16
C1	200	3	A	**16820.15**	7.86	16973.68	0.77	16897.70	92.95	C	**4093.93**	8.72	4114.20	0.70	4104.72	114.55
C1	200	4	A	**16529.97**	6.58	16654.83	0.79	16563.86	95.13	C	**3988.11**	7.89	4019.29	0.76	4009.70	108.95
C1	200	5	A	**17450.12**	6.91	17569.39	0.71	17452.85	108.22	C	**4210.35**	7.58	4214.70	0.70	4210.44	114.11
C1	200	6	A	**17466.85**	8.09	17855.57	0.82	17528.25	114.49	C	4210.35	8.47	4217.46	0.68	4210.35	108.91
C1	200	7	A	**17427.81**	7.80	17867.10	0.84	17539.70	106.81	C	**4191.71**	7.84	4201.95	0.67	4196.00	125.78
C1	200	8	A	**17284.68**	9.03	17477.91	0.77	17366.06	107.13	C	4169.79	7.84	4185.96	0.72	**4165.85**	121.14
C1	200	9	A	17018.38	7.64	17276.58	0.85	16963.26	85.86	C	**4086.01**	7.31	4157.85	0.71	4106.18	131.17
C1	200	10	A	**16749.52**	7.41	17070.56	0.78	16884.73	102.50	C	**4038.56**	8.11	4095.55	0.76	4051.22	102.95
C2	200	1	A	17901.59	6.53	17832.42	0.71	**17787.35**	69.27	C	3547.22	7.75	3635.15	0.94	**3530.70**	82.17
C2	200	2	A	17649.04	6.09	17673.94	0.77	**17591.14**	54.52	C	**3462.97**	7.39	3573.03	0.64	3465.22	70.08
C2	200	3	A	**16984.26**	6.28	17337.81	0.75	17139.14	62.72	C	**3396.56**	7.05	3444.73	0.69	3397.72	75.94
C2	200	4	A	17177.17	5.83	**17034.27**	0.82	17048.28	86.38	C	**3333.64**	6.83	3374.93	0.78	3381.71	89.45
C2	200	5	A	**17567.03**	6.14	17603.86	0.88	17627.05	58.06	C	3496.70	7.14	3568.13	0.79	**3483.03**	67.34
C2	200	6	A	17415.59	6.25	17607.58	0.72	**17321.54**	65.31	C	**3439.13**	7.39	3527.11	0.75	3447.47	74.23
C2	200	7	A	17477.99	6.38	**17419.76**	0.79	17496.18	66.75	C	3491.72	7.11	**3467.02**	0.68	3507.58	67.89
C2	200	8	A	**17177.48**	7.31	17428.72	0.76	17399.34	60.69	C	**3425.01**	7.58	3523.99	0.71	3451.61	69.92
C2	200	9	A	**17478.95**	5.98	17526.67	0.74	17600.02	62.81	C	**3439.75**	7.06	3487.13	0.66	3481.13	78.67
C2	200	10	A	**17024.78**	6.19	17485.71	0.72	17324.16	63.83	C	3408.34	6.92	3499.62	0.73	**3363.75**	68.09
R1	200	1	A	**13897.19**	5.78	14394.10	0.55	13904.64	90.36	C	**5700.76**	6.03	5796.98	0.61	5713.38	100.48
R1	200	2	A	**12367.64**	7.36	12709.44	0.68	12467.81	84.94	C	**4889.25**	7.59	4966.33	0.79	4931.09	85.88
R1	200	3	A	**11122.32**	6.80	11446.22	0.69	11207.92	80.00	C	**4125.01**	7.48	4154.73	0.72	4211.11	85.06
R1	200	4	A	**10388.10**	6.28	10445.86	0.67	10466.36	74.38	C	**3517.42**	7.41	3567.51	0.70	3576.54	93.22
R1	200	5	A	**12294.94**	7.33	12694.58	0.68	12336.34	81.27	C	**4941.12**	7.69	5016.25	0.65	4974.72	89.55
R1	200	6	A	**11332.67**	6.74	11803.95	0.74	11620.63	79.56	C	4340.97	6.94	4396.99	0.73	**4331.41**	80.02
R1	200	7	A	**10669.74**	7.23	10774.63	0.59	10868.82	76.19	C	**3759.99**	8.06	3817.20	0.69	3823.36	80.72
R1	200	8	A	**10216.89**	6.22	10331.22	0.66	10361.97	72.53	C	**3421.79**	7.70	3433.14	0.72	3445.34	81.13
R1	200	9	A	**11639.96**	7.53	11970.78	0.68	11812.63	73.99	C	**4553.55**	7.69	4615.07	0.71	4567.00	77.06
R1	200	10	A	**10780.94**	7.39	10912.28	0.76	10890.60	71.27	C	**3867.58**	7.25	3893.38	0.67	3922.18	75.00
R2	200	1	A	14840.05	6.30	15461.38	0.88	**14355.06**	71.61	C	**4805.18**	6.98	4981.47	0.77	4972.08	52.34
R2	200	2	A	14209.18	6.19	14678.15	0.75	**13543.70**	52.16	C	**4344.74**	6.64	4391.52	0.68	4526.72	63.00
R2	200	3	A	12706.39	6.28	13713.83	0.78	**12103.17**	57.30	C	**3795.58**	6.97	3887.17	0.81	3984.50	61.58
R2	200	4	A	11122.21	6.11	11911.24	0.96	**10520.72**	57.80	C	**3097.62**	6.98	3101.48	0.87	3143.03	72.02
R2	200	5	A	13444.59	6.48	14692.18	0.71	**13081.80**	38.53	C	4375.84	6.77	4521.27	0.68	**4369.26**	54.05
R2	200	6	A	12599.55	6.20	13468.30	0.72	**12543.67**	48.02	C	**3947.06**	7.05	3989.08	0.74	4046.82	49.58
R2	200	7	A	11657.07	6.33	12625.24	0.85	**11137.18**	52.20	C	**3477.60**	6.69	3483.02	0.80	3648.21	58.73
R2	200	8	A	10717.99	6.59	10590.57	0.96	**10274.93**	65.78	C	2896.26	7.23	**2876.54**	0.92	2935.01	73.13
R2	200	9	A	12636.17	6.24	13317.40	1.02	**12586.96**	34.84	C	**4138.39**	7.36	4163.62	0.66	4173.26	44.49
R2	200	10	A	12182.76	6.08	12615.89	0.90	**11347.19**	41.95	C	3817.64	6.42	**3816.96**	0.82	3843.46	36.20
RC1	200	1	A	**10885.35**	6.52	11212.12	0.68	10995.16	94.88	C	4239.69	7.24	4324.98	0.64	**4239.52**	95.44
RC1	200	2	A	**10214.64**	6.94	10513.09	0.66	10413.91	83.36	C	**3882.92**	7.33	3944.67	0.68	3919.43	91.63
RC1	200	3	A	**9644.62**	6.97	9873.20	0.64	9909.46	77.89	C	**3498.29**	7.53	3524.45	0.63	3587.59	81.97
RC1	200	4	A	**9378.79**	6.28	9415.86	0.61	9494.38	74.58	C	3197.80	7.81	**3196.21**	0.71	3289.12	93.98
RC1	200	5	A	**10509.68**	7.41	10743.39	0.69	10603.03	93.73	C	**3984.16**	7.70	4059.55	0.77	4021.30	89.45
RC1	200	6	A	**10368.06**	7.47	10668.14	0.76	10526.77	86.44	C	**3926.56**	8.06	4007.43	0.71	3945.98	96.44
RC1	200	7	A	**10030.10**	7.83	10215.90	0.71	10179.64	83.75	C	**3728.42**	7.25	3802.39	0.73	3799.75	88.03
RC1	200	8	A	**9714.63**	7.42	9895.99	0.65	9856.77	77.39	C	3564.54	7.78	3615.56	0.66	**3552.67**	85.84
RC1	200	9	A	**9737.70**	7.56	9927.84	0.68	9813.89	80.84	C	3541.33	7.47	3588.55	0.69	**3532.69**	85.20
RC1	200	10	A	**9558.28**	7.49	9663.96	0.69	9645.79	74.55	C	3361.24	7.61	3414.21	0.68	**3358.61**	76.81
RC2	200	1	A	**14520.71**	6.09	14781.12	0.73	14529.34	61.31	C	**4114.66**	7.09	4247.83	0.64	4149.91	62.95
RC2	200	2	A	13705.36	5.89	13920.84	0.70	**13425.73**	64.33	C	**3784.42**	7.49	3834.02	0.74	3807.73	67.23
RC2	200	3	A	12827.49	5.53	13459.98	0.78	**11976.30**	64.14	C	**3498.41**	7.03	3551.68	0.72	3701.30	78.53
RC2	200	4	A	11279.77	6.16	12188.24	0.82	**10811.14**	91.09	C	**2980.75**	6.66	3102.24	0.76	3219.43	101.28
RC2	200	5	A	14359.90	5.52	14055.76	0.77	**13392.72**	51.97	C	**3769.78**	6.91	3936.05	0.66	3887.52	51.31
RC2	200	6	A	13565.32	5.63	14031.23	0.61	**13295.09**	45.20	C	**3761.27**	7.31	3833.20	0.71	3820.37	52.39
RC2	200	7	A	12879.30	6.33	13817.98	0.76	**12345.43**	52.80	C	3572.29	7.41	3683.91	0.69	**3545.04**	57.17
RC2	200	8	A	12449.37	5.33	13272.57	0.72	**11989.91**	55.75	C	**3351.74**	7.31	3452.26	0.82	3440.13	54.98
RC2	200	9	A	12109.84	8.09	12509.05	0.98	**11640.05**	41.98	C	**3257.88**	6.84	3418.60	0.79	3326.45	63.41
RC2	200	10	A	11603.96	5.30	12153.31	0.76	**11345.16**	51.19	C	3142.97	6.97	3196.57	0.83	**3005.69**	48.02

				Normal		Quick		MSDAL			Normal		Quick		MSDAL	
Type	Size	No.	Cost	Best	CPU	Best	CPU	Best	CPU	Cost	Best	CPU	Best	CPU	Best	CPU
C1	400	1	A	**37097.84**	13.89	37475.44	1.68	37156.67	368.98	C	**10185.25**	14.16	10220.94	1.67	10186.68	366.30
C1	400	2	A	36456.97	15.70	37022.18	1.83	**36287.65**	375.16	C	10130.08	14.64	10274.94	1.72	**10084.39**	382.86
C1	400	3	A	**35281.26**	14.61	35828.90	1.87	35443.86	366.56	C	9848.58	15.34	10026.01	1.87	**9703.27**	370.94
C1	400	4	A	**33896.21**	15.25	34346.38	1.69	34483.09	353.73	C	**9134.87**	15.75	9302.47	1.84	9149.61	366.09
C1	400	5	A	**37323.54**	14.47	37893.49	1.73	37423.25	375.06	C	**10178.79**	13.89	10222.21	1.54	10179.76	392.80
C1	400	6	A	**37632.95**	16.36	38018.71	1.84	37640.19	389.23	C	10181.25	14.78	10250.33	1.68	**10177.34**	384.77
C1	400	7	A	37656.93	16.64	38019.02	1.76	**37628.29**	368.61	C	10260.60	16.11	10226.32	1.73	**10162.34**	432.42
C1	400	8	A	**37174.88**	18.25	37619.75	1.97	37268.20	356.30	C	10150.06	17.09	10285.09	1.73	**10121.30**	407.55
C1	400	9	A	36532.43	16.80	37112.20	1.87	**36408.62**	371.00	C	9968.79	16.86	10183.87	1.87	**9949.86**	405.22
C1	400	10	A	35847.06	17.11	36605.61	1.83	**35691.67**	354.22	C	9763.37	17.48	10080.98	1.80	**9660.06**	381.63
C2	400	1	A	37088.94	9.75	**36844.04**	1.81	37214.21	305.72	C	**7511.11**	12.08	7628.51	1.49	7574.53	339.64
C2	400	2	A	**34896.10**	11.27	36079.75	1.76	35332.57	296.28	C	**7289.22**	13.11	7371.31	1.92	7507.78	324.50
C2	400	3	A	35503.71	10.48	35763.60	1.85	**35391.65**	296.09	C	**7092.95**	12.33	7250.48	1.68	7460.24	337.22
C2	400	4	A	**34546.07**	10.81	34550.22	1.58	34815.47	358.92	C	**6943.14**	12.81	7137.33	1.64	7172.85	387.73
C2	400	5	A	**36290.36**	10.78	36603.32	1.74	37085.14	297.52	C	**7367.54**	12.70	7605.66	1.74	7493.45	317.08
C2	400	6	A	**35696.75**	11.72	35911.20	1.52	36175.26	291.80	C	**7182.54**	13.78	7453.55	1.70	7331.15	324.31
C2	400	7	A	**36003.90**	11.55	36483.09	1.57	36025.56	304.25	C	**7387.50**	12.80	7558.97	1.49	7407.89	316.39
C2	400	8	A	**35860.75**	10.55	36030.39	1.60	35900.06	307.61	C	**7243.95**	12.75	7351.42	1.75	7317.89	306.81
C2	400	9	A	36265.80	10.64	**35967.77**	1.74	36117.04	320.34	C	**7192.29**	13.11	7424.09	1.56	7395.61	325.31
C2	400	10	A	**35217.67**	10.52	35647.79	1.76	35373.19	321.88	C	7156.69	12.47	7298.86	1.73	**7076.80**	327.58
R1	400	1	A	**28416.94**	10.45	29822.62	1.32	28667.92	329.58	C	**12212.22**	11.89	12758.08	1.34	12254.98	370.05
R1	400	2	A	**25291.12**	13.14	26489.03	1.73	25397.84	318.11	C	10560.12	14.31	10938.76	1.88	**10502.18**	349.16
R1	400	3	A	**22818.20**	12.55	23781.66	1.56	23134.81	297.03	C	**8795.12**	14.61	9167.29	1.70	8956.00	318.00
R1	400	4	A	**21715.80**	12.41	21924.96	1.70	21970.88	303.78	C	**7870.61**	14.59	7997.44	1.60	8043.02	343.14
R1	400	5	A	**25250.04**	13.11	26398.16	1.68	25347.71	317.28	C	10656.56	14.33	11093.36	1.69	**10608.92**	334.73
R1	400	6	A	**23641.90**	15.19	24533.99	1.77	23905.30	301.91	C	**9633.82**	15.55	9964.55	1.64	9713.02	327.27
R1	400	7	A	**22352.49**	13.59	22757.86	1.50	22687.50	301.45	C	**8417.54**	13.22	8551.59	1.63	8555.58	323.72
R1	400	8	A	**21606.92**	13.14	21737.87	1.53	21909.69	298.95	C	**7772.45**	14.66	7878.60	1.63	7905.12	335.55
R1	400	9	A	**24065.69**	13.56	24728.59	1.54	24180.08	297.16	C	**9832.68**	14.64	10109.28	1.56	9921.21	318.06
R1	400	10	A	**22884.25**	13.03	23357.52	1.84	23130.23	285.44	C	**8944.72**	12.83	9194.46	1.65	9078.38	308.25
R2	400	1	A	31463.53	13.16	32393.98	1.63	**30752.74**	321.22	C	**10370.89**	13.03	10717.94	1.49	10799.49	185.02
R2	400	2	A	30025.97	12.03	30857.05	1.35	**27372.86**	220.03	C	**9171.98**	12.80	9547.03	1.53	9635.11	267.55
R2	400	3	A	27621.15	11.81	28633.30	1.59	**24922.94**	264.44	C	**8089.29**	11.75	8290.23	1.47	8338.94	330.72
R2	400	4	A	21894.05	10.66	24919.20	1.54	**21552.34**	331.88	C	**6493.74**	13.39	6724.16	1.64	6809.50	306.08
R2	400	5	A	30007.57	11.42	30719.03	1.71	**27302.03**	179.00	C	**9447.63**	12.55	9712.17	1.49	9644.09	204.39
R2	400	6	A	26785.61	11.17	27992.37	1.35	**24400.15**	183.64	C	**8528.13**	12.92	8855.36	1.79	9092.40	240.33
R2	400	7	A	24206.06	10.49	27253.41	1.74	**22749.95**	248.61	C	**7449.61**	12.17	7747.22	1.59	8004.41	269.05
R2	400	8	A	**20392.97**	10.58	24019.43	1.91	21298.98	362.19	C	**6176.44**	14.64	6475.11	1.59	6552.64	313.17
R2	400	9	A	27132.32	11.38	29681.60	1.46	**25891.73**	155.73	C	**8864.43**	12.95	9237.98	1.63	8991.19	196.50
R2	400	10	A	25154.26	12.39	26245.48	2.23	**23894.38**	184.95	C	**8386.89**	13.20	8763.91	1.62	8567.66	200.52
RC1	400	1	A	**23429.93**	12.98	24268.90	1.74	23667.18	348.45	C	**9812.02**	14.42	10188.93	1.55	9839.00	360.55
RC1	400	2	A	**21860.59**	13.50	22547.76	1.66	22066.80	327.91	C	**8859.79**	15.70	9185.34	1.74	8970.15	349.41
RC1	400	3	A	**20446.10**	12.78	20808.42	1.66	20597.40	316.80	C	**8089.51**	15.17	8259.40	1.79	8140.88	337.50
RC1	400	4	A	**19809.80**	13.38	19989.00	1.53	20080.59	331.70	C	**7558.39**	14.28	7633.66	1.51	7723.72	361.88
RC1	400	5	A	**22197.31**	14.45	23174.23	1.80	22271.61	341.41	C	**9147.86**	15.50	9605.65	1.60	9306.67	341.63
RC1	400	6	A	**21760.79**	14.47	22966.89	1.51	22257.21	338.55	C	**9127.43**	15.36	9409.75	1.60	9150.17	338.03
RC1	400	7	A	**21460.28**	15.36	22540.43	1.70	21663.84	327.77	C	**8754.82**	13.91	9173.01	1.42	8844.62	328.13
RC1	400	8	A	**20879.27**	14.20	21637.60	1.67	21004.48	313.55	C	8427.69	14.56	8704.36	1.63	**8391.29**	323.69
RC1	400	9	A	**20742.67**	15.06	21077.78	1.75	20841.74	311.05	C	8318.95	13.50	8629.20	1.59	**8297.20**	309.86
RC1	400	10	A	**20197.91**	12.97	20941.87	1.60	20439.53	306.94	C	**8025.24**	13.89	8200.34	1.73	8085.39	320.97
RC2	400	1	A	**29970.85**	12.50	31083.21	1.67	30354.46	348.42	C	**8849.39**	11.27	9210.51	1.63	9035.96	267.58
RC2	400	2	A	28733.62	11.44	28600.57	1.59	**26847.73**	341.09	C	**7984.16**	11.16	8150.01	1.46	8567.81	356.14
RC2	400	3	A	26844.11	10.47	26876.59	1.43	**24497.60**	345.91	C	**7137.48**	12.00	7312.11	1.32	7763.58	361.31
RC2	400	4	A	**20679.44**	14.63	23864.14	1.55	21047.48	359.88	C	**5769.86**	13.31	6012.29	1.71	6609.23	375.17
RC2	400	5	A	28456.96	11.42	29334.89	1.74	**28217.45**	264.09	C	**8142.32**	12.47	8509.85	1.70	8289.54	272.08
RC2	400	6	A	28627.03	11.56	28773.46	1.50	**27157.56**	292.84	C	**8022.63**	11.59	8345.58	1.76	8063.89	302.53
RC2	400	7	A	27342.13	11.20	27919.21	1.46	**26655.32**	247.58	C	**7589.56**	12.14	7937.47	1.38	7724.78	259.39
RC2	400	8	A	25521.11	12.03	26044.41	1.60	**25030.18**	250.05	C	**7197.08**	13.39	7577.69	1.21	7338.61	257.86
RC2	400	9	A	**23348.97**	13.08	26031.85	1.28	23893.36	255.91	C	**7003.77**	13.14	7380.02	1.73	7235.74	287.03
RC2	400	10	A	**22963.14**	16.47	25173.19	1.73	23346.33	231.13	C	**6567.66**	13.45	6981.33	1.35	6931.63	258.11

				Normal		Quick		MSDAL			Normal		Quick		MSDAL	
Type	Size	No.	Cost	Best	CPU	Best	CPU	Best	CPU	Cost	Best	CPU	Best	CPU	Best	CPU
C1	600	1	A	**59586.47**	21.09	60251.16	2.57	59689.55	760.17	C	18716.49	19.03	18772.28	2.62	**18707.09**	793.52
C1	600	2	A	**58571.76**	24.34	59584.03	3.14	58755.09	785.00	C	**18502.45**	22.34	18816.38	2.97	18526.04	793.52
C1	600	3	A	**56829.20**	24.06	58337.92	3.05	57563.30	771.61	C	17970.54	22.67	18428.04	3.21	**17911.72**	799.81
C1	600	4	A	**55567.63**	22.53	56435.78	3.17	56426.93	783.73	C	**17123.40**	22.58	17867.06	3.13	17264.42	779.80
C1	600	5	A	59943.50	21.64	60568.12	2.91	**59855.49**	757.41	C	**18693.84**	17.95	18816.36	2.61	18706.06	822.45
C1	600	6	A	**60318.27**	23.66	61257.29	3.39	60653.99	801.13	C	**18707.56**	21.69	18790.89	2.97	18717.27	813.41
C1	600	7	A	60294.54	24.36	61187.41	3.03	**60286.25**	776.58	C	**18701.59**	21.66	18836.16	2.75	18704.18	863.47
C1	600	8	A	**59595.55**	25.70	60650.70	3.27	59920.91	776.97	C	18682.18	23.31	18930.05	2.94	**18571.11**	801.09
C1	600	9	A	58567.22	27.03	59959.39	2.91	**58156.55**	786.81	C	18242.30	23.27	18902.90	2.79	**17887.99**	815.83
C1	600	10	A	57896.52	27.27	59034.15	3.42	**57870.92**	754.78	C	17887.74	24.09	18456.34	2.85	**17639.89**	815.38
C2	600	1	A	58082.40	17.44	**57778.66**	2.98	58825.82	714.53	C	13332.62	15.05	13633.65	2.78	**13015.11**	716.86
C2	600	2	A	57698.77	19.27	**57455.98**	3.29	58081.86	741.48	C	**12918.37**	18.03	13436.82	2.83	13020.56	742.78
C2	600	3	A	**55292.05**	16.86	56498.89	3.04	55943.84	775.75	C	**12715.31**	16.64	13383.17	2.53	12849.37	779.22
C2	600	4	A	**54421.80**	16.83	55239.27	2.66	55141.06	863.83	C	**12377.14**	16.34	12616.88	2.73	12532.26	830.28
C2	600	5	A	**57227.76**	19.17	57276.94	2.96	58023.94	714.28	C	**13080.26**	17.11	13722.44	2.66	13424.80	703.59
C2	600	6	A	**56603.26**	17.83	57771.32	3.13	57196.13	738.19	C	**13050.43**	17.91	13128.38	2.93	13769.02	745.63
C2	600	7	A	**56829.07**	19.36	57451.41	2.83	57811.18	739.98	C	**13223.82**	17.24	13522.86	2.68	13399.36	711.77
C2	600	8	A	**56119.74**	18.13	57035.73	2.85	56595.38	776.39	C	**12680.30**	17.27	13306.05	2.64	13038.78	764.05
C2	600	9	A	**56183.81**	19.19	56817.45	2.59	56210.82	761.52	C	**12995.09**	17.88	13634.73	2.83	13054.19	754.39
C2	600	10	A	55607.16	18.30	**55355.08**	2.68	55862.66	735.16	C	**12730.64**	16.95	12903.33	2.82	13189.45	753.11
R1	600	1	A	**48035.25**	16.31	52429.27	2.22	48768.16	724.11	C	**24277.93**	15.97	25960.30	2.29	24292.87	751.25
R1	600	2	A	**42172.33**	21.94	45845.44	2.55	43036.65	683.81	C	**20575.01**	20.86	22049.80	2.95	20600.75	697.09
R1	600	3	A	**39089.63**	21.48	41272.03	2.64	39796.18	668.28	C	**17908.61**	20.52	18841.63	2.70	18382.28	691.66
R1	600	4	A	**36769.21**	20.94	37553.14	2.61	37493.03	683.09	C	**15765.01**	19.45	16359.93	2.60	16321.85	718.95
R1	600	5	A	**42517.05**	23.97	45412.47	2.81	43303.32	671.16	C	**21118.63**	18.58	22285.62	2.50	21421.98	674.55
R1	600	6	A	**40314.96**	21.25	42180.38	2.72	41316.95	658.94	C	**19015.45**	20.27	19904.84	2.79	19323.65	672.50
R1	600	7	A	**38234.20**	20.67	39300.59	2.95	38989.63	672.41	C	**17025.78**	20.81	17719.99	2.86	17921.75	681.05
R1	600	8	A	**36390.44**	21.97	36895.36	2.83	36982.48	694.75	C	**15616.18**	21.83	15960.56	2.60	16051.66	722.38
R1	600	9	A	**40960.30**	23.81	42846.98	2.57	41915.91	654.11	C	**19767.32**	20.03	20940.92	2.70	20085.27	656.09
R1	600	10	A	**39521.70**	23.38	40885.47	2.88	40068.23	637.75	C	**18449.85**	19.89	19364.35	2.98	18891.04	652.08
R2	600	1	A	**53192.13**	20.61	54553.67	3.14	55926.79	826.92	C	**19697.79**	17.81	20677.15	2.78	20981.80	470.61
R2	600	2	A	49406.02	18.70	51133.11	2.41	**48401.15**	473.86	C	**17182.59**	17.03	18136.44	2.82	18525.78	562.38
R2	600	3	A	44327.34	17.02	47184.13	2.50	**40919.19**	586.42	C	**14510.42**	17.45	15213.28	2.80	16550.00	616.45
R2	600	4	A	**35083.09**	17.39	37559.61	2.66	35706.92	678.59	C	**11449.41**	17.55	11981.02	2.44	13157.25	707.73
R2	600	5	A	50693.53	20.89	52040.10	2.77	**49199.77**	372.17	C	**18294.03**	17.56	19314.34	2.56	19533.05	447.58
R2	600	6	A	46572.58	17.70	49309.62	2.34	**42134.19**	405.91	C	**16356.36**	17.83	17009.98	2.54	18749.86	404.58
R2	600	7	A	42750.32	16.58	45061.93	2.37	**39024.05**	526.75	C	**13805.47**	15.74	14368.23	2.75	14898.00	544.27
R2	600	8	A	**34233.23**	17.80	36432.52	2.90	35947.38	688.89	C	**10865.64**	19.80	11424.00	2.54	13082.88	702.95
R2	600	9	A	49347.98	19.33	50573.58	2.63	**45496.03**	447.63	C	**17034.98**	17.20	18419.94	2.82	18567.27	469.55
R2	600	10	A	42719.16	19.41	47946.27	2.81	**42371.50**	437.70	C	**16213.20**	18.38	17308.08	2.83	17087.31	521.34
RC1	600	1	A	**38568.22**	21.20	40651.50	2.54	38981.34	723.89	C	**18469.16**	19.52	19527.59	2.69	18691.07	742.28
RC1	600	2	A	**36115.10**	22.45	37995.47	2.51	36931.09	701.08	C	**16864.48**	20.17	17733.59	2.70	17192.16	703.69
RC1	600	3	A	**34183.42**	22.06	35463.35	2.62	35118.63	694.70	C	**15442.62**	19.81	15921.81	2.99	15987.36	703.31
RC1	600	4	A	**33177.38**	22.61	33671.32	2.65	34036.86	714.39	C	**14541.66**	20.72	14912.05	2.72	15029.87	729.91
RC1	600	5	A	**37031.67**	21.27	39463.80	2.79	37890.20	698.89	C	**17505.80**	20.86	18765.61	2.64	18122.45	700.42
RC1	600	6	A	**36479.06**	24.06	38764.59	2.74	36917.19	688.64	C	**17359.22**	18.42	18481.13	2.54	17417.86	692.80
RC1	600	7	A	**35613.49**	23.75	37493.77	2.72	36045.20	673.61	C	**16801.52**	20.08	17761.39	2.64	16906.27	687.48
RC1	600	8	A	**34989.07**	21.52	36606.20	2.54	35731.11	658.69	C	**16303.91**	20.06	17191.34	2.40	16665.67	667.88
RC1	600	9	A	**34969.48**	24.63	36047.57	2.61	35355.26	661.61	C	**16224.11**	19.67	16950.09	2.72	16387.88	667.16
RC1	600	10	A	**34707.66**	22.63	35576.92	2.65	35048.50	647.77	C	**15828.02**	21.30	16693.23	2.52	16380.54	666.22
RC2	600	1	A	**49713.11**	20.45	51764.00	2.31	50965.70	728.20	C	**16473.45**	16.48	17478.37	2.42	16887.29	517.75
RC2	600	2	A	46584.05	18.78	48268.59	2.34	**45977.09**	669.75	C	**14470.66**	15.52	15466.53	2.45	15607.40	611.44
RC2	600	3	A	40861.20	15.86	43593.14	2.12	**40652.52**	675.24	C	**12795.89**	17.36	13336.19	2.37	14175.00	716.31
RC2	600	4	A	34942.97	18.14	36235.98	3.93	**34595.77**	711.36	C	**10402.35**	18.97	10977.49	2.74	11806.37	749.49
RC2	600	5	A	47600.73	18.58	49351.70	2.32	**46462.01**	552.95	C	**15449.32**	16.30	16498.27	2.59	16152.60	525.72
RC2	600	6	A	**47295.00**	18.53	48880.58	2.46	47640.27	589.58	C	**15413.57**	17.66	16411.56	2.51	15578.28	581.92
RC2	600	7	A	**44216.27**	18.97	48061.45	2.85	44631.66	655.25	C	**14576.06**	17.66	15692.77	2.68	15535.85	542.03
RC2	600	8	A	42675.06	17.70	45288.27	2.47	**41694.33**	487.59	C	**14095.21**	18.23	15109.38	2.43	14289.59	536.38
RC2	600	9	A	**39103.91**	17.44	44089.67	2.43	40365.67	514.88	C	**13627.13**	16.22	14472.70	2.90	14628.42	614.84
RC2	600	10	A	**37492.89**	18.00	42200.48	2.86	38217.84	485.91	C	**13019.75**	17.02	13427.16	2.80	13745.89	578.67

				Normal		Quick		MSDAL			Normal		Quick		MSDAL	
Type	Size	No.	Cost	Best	CPU	Best	CPU	Best	CPU	Cost	Best	CPU	Best	CPU	Best	CPU
C1	800	1	A	**85377.70**	24.34	87249.03	3.68	85904.80	1331.39	C	**31306.21**	26.94	31523.92	3.90	31846.67	1306.95
C1	800	2	A	**83500.72**	30.70	85263.26	4.62	84580.95	1330.84	C	**30995.73**	32.52	31555.57	4.29	31581.31	1348.14
C1	800	3	A	**80226.37**	29.25	82115.45	4.53	80617.52	1296.75	C	29599.34	33.39	30971.64	4.51	**29319.89**	1433.02
C1	800	4	A	**77236.34**	31.89	79159.29	4.07	77870.83	1309.70	C	**27715.07**	33.94	28990.33	4.38	27860.33	1410.50
C1	800	5	A	**85756.73**	33.09	87571.44	4.68	86500.22	1329.36	C	**31268.45**	31.39	31478.48	4.37	31521.56	1433.78
C1	800	6	A	86476.10	32.55	88016.56	4.79	**86386.20**	1319.61	C	**31312.16**	31.77	31459.48	3.86	31456.02	1448.78
C1	800	7	A	86513.45	33.97	88049.80	4.53	**86215.78**	1350.13	C	**31284.95**	30.66	31682.92	4.02	31456.56	1478.83
C1	800	8	A	**85234.16**	35.94	86863.92	4.77	86156.11	1319.23	C	**31179.57**	34.95	31566.81	4.11	31424.64	1447.25
C1	800	9	A	**83810.09**	36.45	85640.44	5.10	84033.43	1313.17	C	30582.53	35.94	31609.20	4.23	**30327.43**	1438.94
C1	800	10	A	**82036.56**	37.22	83836.70	4.78	82053.96	1331.73	C	29925.15	34.31	30988.98	4.61	**29490.43**	1418.89
C2	800	1	A	**78151.77**	24.05	78795.09	4.56	81655.68	1297.78	C	19104.26	25.02	20023.77	4.43	**18867.04**	1433.24
C2	800	2	A	**76323.83**	24.59	76371.98	4.65	79202.33	1341.41	C	**19022.27**	25.55	20058.74	4.40	19618.88	1445.19
C2	800	3	A	**76453.56**	24.36	76669.95	4.70	78244.81	1319.11	C	**18778.22**	24.98	19788.18	4.09	19866.43	1533.70
C2	800	4	A	**73761.73**	23.89	74707.88	3.51	74952.77	1458.69	C	**18312.74**	24.53	18730.22	3.81	19095.48	1621.47
C2	800	5	A	**77151.81**	25.80	79037.20	4.86	80079.09	1291.39	C	19142.23	23.69	20286.71	4.11	**18951.86**	1323.30
C2	800	6	A	77715.28	25.59	**77264.73**	4.43	78828.71	1308.94	C	18883.89	26.00	19710.77	4.40	**18824.79**	1250.08
C2	800	7	A	**77071.63**	25.75	77704.16	3.93	80072.08	1317.11	C	**19662.82**	26.00	20344.91	3.50	19959.09	1278.97
C2	800	8	A	**76148.19**	23.06	77131.97	3.93	78157.06	1336.98	C	**18870.43**	26.09	19077.93	4.45	19567.56	1278.56
C2	800	9	A	76617.84	24.99	**75729.57**	4.07	78640.20	1339.33	C	**19153.84**	25.03	19610.47	3.81	19663.36	1302.88
C2	800	10	A	**75010.33**	25.22	75650.88	3.65	76765.70	1325.14	C	**18528.86**	24.08	18788.49	4.01	19182.40	1297.06
R1	800	1	A	**72590.84**	21.42	80142.40	3.39	73423.38	1262.56	C	**40722.32**	24.86	45003.90	3.77	40738.97	1405.81
R1	800	2	A	**64562.15**	28.84	71649.59	4.42	65257.05	1202.75	C	**35039.90**	28.08	38795.79	4.04	35459.84	1326.39
R1	800	3	A	**57934.34**	28.78	62320.81	3.97	59820.10	1182.33	C	**29859.11**	28.67	31947.83	4.12	30784.69	1323.24
R1	800	4	A	**54311.52**	31.05	55525.06	3.96	55631.60	1226.55	C	**26571.09**	33.17	27565.42	4.10	27472.15	1375.56
R1	800	5	A	**64073.48**	28.22	69141.66	4.30	65112.59	1170.44	C	**35683.17**	29.23	38383.95	4.07	36029.45	1318.28
R1	800	6	A	**59854.79**	29.19	63223.56	3.99	61688.93	1162.14	C	**31401.22**	33.06	33955.09	4.27	32471.26	1309.08
R1	800	7	A	**56186.56**	33.22	58015.08	4.32	57701.16	1170.03	C	**28486.17**	31.64	29910.75	3.74	29303.08	1351.52
R1	800	8	A	**54076.86**	29.63	54757.41	4.24	55067.66	1240.86	C	**26306.48**	31.56	27135.59	4.10	27181.22	1423.31
R1	800	9	A	**61284.97**	29.66	64894.08	4.27	63014.13	1158.69	C	**33099.81**	28.00	35553.95	3.86	33991.16	1292.28
R1	800	10	A	**59296.93**	33.11	61743.25	3.77	61051.45	1143.00	C	**31285.10**	31.86	33166.25	4.14	31979.75	1272.61
R2	800	1	A	76800.86	29.66	79517.78	4.19	**76129.39**	1142.08	C	**31588.51**	26.33	33559.02	3.99	33865.73	925.58
R2	800	2	A	69237.10	27.05	73591.16	3.47	**68541.41**	872.50	C	**27601.50**	25.88	29567.79	3.80	29710.76	1113.41
R2	800	3	A	**60335.26**	23.97	64950.06	3.62	60673.82	1112.23	C	**23026.31**	25.31	24313.38	3.68	25814.85	1246.00
R2	800	4	A	**52007.50**	22.39	53861.32	4.12	52897.81	1199.94	C	**18353.81**	25.31	19004.66	3.79	19584.68	1263.48
R2	800	5	A	73100.38	31.02	75559.52	3.60	**70570.70**	1043.31	C	**29379.08**	25.31	31689.44	4.14	31565.70	830.22
R2	800	6	A	66345.27	24.20	70034.84	4.15	**63607.13**	844.80	C	**25993.16**	25.08	27656.26	4.07	28039.16	1109.69
R2	800	7	A	**59135.41**	25.00	63529.04	4.16	59941.44	1180.83	C	**22317.59**	24.44	23098.38	3.94	25839.81	1042.83
R2	800	8	A	**49862.94**	25.02	53609.49	4.03	50457.42	1234.20	C	**17544.24**	23.49	18497.66	4.03	19474.12	1223.72
R2	800	9	A	70365.53	27.66	72567.06	4.63	**67169.72**	783.03	C	**27794.89**	23.05	29961.84	4.46	30039.87	763.03
R2	800	10	A	65480.44	26.00	68886.14	4.03	**64422.00**	634.05	C	**26502.74**	24.23	28330.69	4.11	28481.98	883.42
RC1	800	1	A	**58036.37**	28.78	62015.59	3.77	59150.52	1237.28	C	**26982.44**	31.91	28645.85	4.02	27446.71	1191.06
RC1	800	2	A	**53906.74**	31.30	57340.22	4.12	55364.17	1204.48	C	**31311.17**	32.22	33723.21	4.32	32201.65	1253.56
RC1	800	3	A	**51423.03**	31.91	54110.52	3.90	52680.52	1231.88	C	**28263.09**	30.61	30767.61	4.16	29402.58	1234.13
RC1	800	4	A	**49389.16**	30.49	50706.70	4.05	50474.12	1265.33	C	**26776.21**	32.92	28195.53	4.11	27766.02	1252.24
RC1	800	5	A	**55811.28**	29.91	59848.09	4.21	57066.52	1202.56	C	**24802.80**	30.33	25650.30	4.47	25379.72	1285.17
RC1	800	6	A	**55368.05**	31.22	58474.58	4.64	55904.77	1207.13	C	**30023.54**	32.64	32207.36	3.95	30521.64	1225.08
RC1	800	7	A	**53951.41**	31.86	57550.13	3.90	54825.86	1186.95	C	**29376.62**	33.39	31796.89	3.94	29693.74	1213.64
RC1	800	8	A	**52573.27**	29.80	55241.32	4.45	54077.31	1172.06	C	**28469.93**	30.66	30734.35	4.31	29040.12	1201.38
RC1	800	9	A	**52536.89**	31.53	55689.27	3.87	54060.53	1164.98	C	**27713.70**	34.05	29598.43	4.29	28744.38	1194.99
RC1	800	10	A	**52055.42**	30.16	53608.05	3.94	52977.40	1158.20	C	**27503.60**	30.81	29485.62	4.07	28365.75	1190.58
RC2	800	1	A	**68932.23**	29.66	72477.63	4.33	72564.38	1180.42	C	**25510.42**	25.92	27796.10	3.67	26967.79	1094.53
RC2	800	2	A	**63429.73**	23.53	67557.63	3.54	65076.57	1235.95	C	**22564.77**	23.78	24898.82	3.49	24967.48	1241.05
RC2	800	3	A	**56750.79**	23.84	60341.63	3.78	57898.17	1325.23	C	**19486.10**	24.56	20693.95	3.58	21165.76	1392.27
RC2	800	4	A	**49084.96**	22.77	52338.15	3.82	51738.96	1349.83	C	**15650.35**	24.39	16800.43	4.37	17256.03	1489.83
RC2	800	5	A	65449.16	25.58	70432.32	4.10	**65302.06**	1051.83	C	**23440.24**	24.03	26135.93	3.92	24480.05	1150.34
RC2	800	6	A	**62731.40**	23.25	68276.70	3.98	65842.89	1086.66	C	**23389.50**	24.44	25896.43	3.91	24864.24	1038.98
RC2	800	7	A	**61200.18**	25.38	65939.22	4.00	63694.95	1030.31	C	**22536.80**	27.53	24206.02	4.03	23640.37	1088.58
RC2	800	8	A	**57763.84**	22.97	64114.81	4.03	59340.22	977.31	C	**21481.50**	24.59	23574.21	3.58	22731.98	1052.16
RC2	800	9	A	**56770.02**	23.45	63392.97	3.76	57960.22	995.58	C	**20964.52**	23.45	22809.26	3.72	22423.33	1115.30
RC2	800	10	A	**54190.70**	22.80	61164.90	4.30	54409.41	998.69	C	**20208.70**	25.58	22144.52	3.50	21656.00	1091.56

				Normal		Quick		MSDAL			Normal		Quick		MSDAL	
Type	Size	No.	Cost	Best	CPU	Best	CPU	Best	CPU	Cost	Best	CPU	Best	CPU	Best	CPU
C1	1000	1	A	**117485.38**	35.27	120347.76	5.91	118787.14	1983.70	C	**50065.08**	36.55	50238.08	6.44	50787.86	2021.80
C1	1000	2	A	**115410.43**	46.91	117360.13	7.32	116725.88	1987.66	C	**49606.12**	45.03	50419.34	6.77	49784.06	2048.88
C1	1000	3	A	**111346.16**	47.77	114048.98	7.12	112483.80	1981.67	C	47948.33	49.47	49668.22	6.61	**47210.25**	2058.17
C1	1000	4	A	**104794.52**	44.94	107387.16	6.71	106391.73	2042.83	C	**43895.81**	48.11	45264.98	6.87	43981.17	2177.30
C1	1000	5	A	**117881.98**	42.61	120398.51	7.08	118762.64	2028.78	C	**50075.16**	45.20	50362.09	6.42	50310.32	2201.94
C1	1000	6	A	**118269.16**	46.00	121323.17	7.04	120062.98	2027.02	C	**50131.77**	42.36	50444.50	6.26	50139.08	2208.06
C1	1000	7	A	119287.70	43.56	121974.28	6.75	**118882.44**	2044.41	C	**50179.69**	43.84	51141.12	6.44	50344.85	2201.61
C1	1000	8	A	**117465.45**	47.41	120240.02	7.31	118720.60	2010.92	C	50078.75	44.41	50381.91	6.63	**50059.20**	2206.91
C1	1000	9	A	**114729.25**	49.77	118335.72	7.17	115666.78	2043.36	C	49033.63	45.05	50603.65	6.28	**48281.63**	2239.78
C1	1000	10	A	**113565.70**	48.25	116745.81	7.00	113723.05	2018.86	C	48006.15	48.13	49568.78	6.83	**47767.31**	2189.56
C2	1000	1	A	**101055.51**	33.80	101623.29	7.08	104405.41	1895.28	C	**26914.42**	33.97	27008.98	6.90	27565.94	2273.83
C2	1000	2	A	**99865.72**	35.19	101510.52	6.25	104122.18	1910.50	C	**26360.07**	34.03	27583.53	6.43	27451.23	2237.17
C2	1000	3	A	**97243.95**	32.99	99517.79	6.27	100344.77	1969.92	C	**26335.14**	33.88	27823.71	5.77	27775.83	2340.63
C2	1000	4	A	**95365.89**	32.52	97364.61	6.97	99061.77	2187.19	C	**25404.70**	35.55	26633.39	6.70	26171.65	2457.75
C2	1000	5	A	**99935.95**	34.17	100606.26	7.01	102803.17	1850.39	C	**26709.19**	35.14	27834.70	6.23	27069.58	2163.28
C2	1000	6	A	**99364.25**	32.88	100004.38	6.19	102681.81	1896.06	C	**26926.17**	31.20	27062.52	6.27	27874.57	2252.98
C2	1000	7	A	**99228.42**	33.66	100757.37	6.29	103154.60	1881.92	C	**26664.01**	32.72	27830.90	6.30	27906.83	2217.74
C2	1000	8	A	**97999.53**	34.11	98770.97	5.60	101995.64	1915.75	C	**26398.24**	34.13	26721.31	6.31	27492.01	2220.94
C2	1000	9	A	**98829.27**	33.20	98970.20	6.57	101620.37	1898.66	C	**26563.02**	33.48	27419.99	6.08	27589.01	2241.56
C2	1000	10	A	**96876.09**	32.73	96876.20	6.61	101197.13	1938.42	C	**26151.47**	32.24	26309.42	6.72	26891.55	2240.75
R1	1000	1	A	**98182.30**	30.14	109664.48	5.48	100212.48	2040.98	C	**58614.10**	28.39	64895.92	5.90	58943.11	2178.30
R1	1000	2	A	**87719.66**	39.27	97165.38	6.27	89838.07	1853.09	C	**50395.03**	42.08	56256.66	7.05	51655.98	2084.27
R1	1000	3	A	**79314.20**	40.33	85850.81	6.15	81944.28	1725.16	C	**43793.72**	40.02	47288.29	6.68	45856.57	2062.03
R1	1000	4	A	**74883.88**	40.28	76806.78	6.09	76926.73	1806.22	C	**39729.29**	41.39	41441.66	6.26	41019.65	2139.47
R1	1000	5	A	**87170.08**	36.02	96289.09	6.12	90778.97	1703.67	C	**51998.27**	40.47	56601.41	6.74	53446.32	2078.80
R1	1000	6	A	**82132.25**	39.27	87260.02	6.45	85998.64	1694.72	C	**46956.65**	42.64	50449.98	5.72	48283.29	2045.13
R1	1000	7	A	**77698.93**	40.56	80861.40	6.58	80195.64	1733.50	C	**42431.94**	40.38	44954.71	6.59	44417.54	2087.34
R1	1000	8	A	**74621.00**	43.72	75855.92	6.72	76503.94	1761.13	C	**39378.73**	44.86	40619.76	6.92	40368.48	2159.92
R1	1000	9	A	**85080.58**	37.31	91304.04	6.58	88163.81	1688.74	C	**49370.69**	41.14	53320.46	6.52	50974.68	2044.73
R1	1000	10	A	**82393.37**	39.45	85552.00	6.33	85534.08	1671.95	C	**46699.48**	42.25	50170.77	6.33	48615.96	2017.91
R2	1000	1	A	**104301.16**	42.06	109797.56	6.40	110184.33	1313.19	C	**45786.67**	35.08	50053.20	6.15	49601.56	1829.30
R2	1000	2	A	**94750.47**	38.34	98699.40	6.21	97442.41	1501.20	C	**39158.92**	33.06	43441.53	5.86	45550.71	1918.45
R2	1000	3	A	**80347.20**	38.19	88041.63	6.37	83346.44	1714.33	C	**31949.84**	30.06	35257.91	6.30	37908.70	2233.11
R2	1000	4	A	**65589.11**	28.13	71404.65	5.94	70656.05	1757.58	C	**24545.11**	33.19	25538.90	6.14	28511.63	2028.28
R2	1000	5	A	**100037.95**	42.27	105872.91	6.47	104693.34	1206.02	C	**43108.04**	34.14	46894.20	5.72	46757.77	1514.56
R2	1000	6	A	**87520.57**	35.44	96335.49	6.32	90800.22	1400.36	C	**37719.80**	30.58	41219.29	6.29	42134.09	1608.91
R2	1000	7	A	**75569.19**	34.63	84246.91	6.02	76471.25	1706.22	C	**30835.81**	33.17	33501.52	5.41	35172.42	1887.48
R2	1000	8	A	**63537.16**	31.73	68010.58	6.49	69906.48	1778.36	C	**23537.39**	32.70	24771.12	6.22	27657.81	2122.02
R2	1000	9	A	97251.31	40.09	102823.86	6.98	**96569.63**	1050.55	C	**40990.00**	34.45	45290.50	6.40	44953.75	1488.67
R2	1000	10	A	93063.65	37.48	97999.03	6.01	**90303.81**	1071.19	C	**39200.86**	35.94	43047.27	6.15	43142.36	1199.31
RC1	1000	1	A	**79959.62**	39.94	87124.06	6.61	82815.73	1876.09	C	**47049.28**	39.42	51630.96	6.20	48849.41	2037.16
RC1	1000	2	A	**73959.01**	42.27	80266.86	6.58	77839.84	1885.22	C	**42589.15**	42.98	47411.96	7.06	44660.72	2122.11
RC1	1000	3	A	**70309.67**	44.02	74201.64	6.45	72778.05	1865.02	C	**39278.03**	41.47	42734.00	7.31	41370.69	2157.48
RC1	1000	4	A	**68081.25**	41.99	70066.24	6.05	69462.56	1937.08	C	**37534.31**	43.00	39388.42	6.13	38606.30	2212.52
RC1	1000	5	A	**76776.20**	40.53	83597.54	6.38	79230.56	1877.63	C	**44786.32**	40.30	49698.38	7.04	46503.62	2113.03
RC1	1000	6	A	**75310.20**	42.14	81967.75	5.89	77781.09	1877.72	C	**44116.07**	40.05	47728.04	6.17	46316.18	2081.97
RC1	1000	7	A	**73748.70**	44.55	80043.14	5.73	75976.66	1845.33	C	**43211.56**	43.08	47581.91	6.12	45117.43	2072.89
RC1	1000	8	A	**72488.42**	40.33	76945.84	5.87	75322.36	1837.09	C	**41422.41**	40.25	44639.06	6.40	44002.29	2058.11
RC1	1000	9	A	**72499.64**	40.98	76907.17	5.79	74556.65	1833.77	C	**41758.04**	40.00	44926.40	6.26	43665.91	2002.34
RC1	1000	10	A	**71752.19**	41.52	74512.42	6.53	74562.46	1826.00	C	**40857.08**	39.42	43451.27	6.81	42647.78	1875.02
RC2	1000	1	A	**92100.95**	36.59	96764.25	6.09	96431.08	1718.86	C	**36591.16**	36.44	41577.72	5.55	39684.66	1405.31
RC2	1000	2	A	**82790.18**	33.86	88165.16	5.85	87586.31	1651.05	C	**32209.94**	31.31	35873.65	5.27	36250.77	1799.69
RC2	1000	3	A	**69548.62**	25.39	78798.52	5.45	74445.37	1918.56	C	**26761.99**	32.25	29919.70	6.09	30443.51	2185.44
RC2	1000	4	A	**61890.14**	33.45	67857.64	5.98	64687.95	2113.05	C	**21596.82**	36.56	24407.51	5.93	25704.52	2427.97
RC2	1000	5	A	**87244.16**	36.31	94271.84	6.03	88789.04	1470.70	C	**34349.43**	36.20	39046.23	5.33	36891.79	1587.11
RC2	1000	6	A	**85087.14**	35.89	93215.06	5.53	90015.29	1518.77	C	**34312.05**	32.77	38565.73	6.29	37837.96	1567.25
RC2	1000	7	A	**83513.57**	34.17	91773.47	6.06	86259.89	1397.72	C	**33036.03**	33.45	37391.91	5.69	36658.46	1595.92
RC2	1000	8	A	80933.86	31.77	89062.81	6.07	**80197.05**	1495.63	C	**31815.12**	33.99	35265.32	5.70	33952.09	1539.23
RC2	1000	9	A	**73604.45**	33.33	86918.95	5.83	79337.48	1414.70	C	**30717.02**	31.81	35010.23	5.95	33757.40	1591.59
RC2	1000	10	A	**71194.46**	33.72	83200.52	5.99	75617.04	1443.11	C	**29522.06**	34.28	33437.45	5.66	33072.73	1567.84

References

Bräysy, O. (2003). A reactive variable neighborhood search for the vehicle routing problem with time windows. *INFORMS Journal of Computing, 15*(4), 347–368.

Bräysy, O., Gendreau, M., Hasle, G., & Løkketangen, A. (2007a). *A survey of heuristics for the vehicle routing problem, Part I: Basic problems and supply side extensions*. Working paper, SINTEF ICT, Norway.

Bräysy, O., Gendreau, M., Hasle, G., & Løkketangen, A. (2007b). *A survey of heuristics for the vehicle routing problem, Part II: Demand side extensions*. Working paper, SINTEF ICT, Norway.

Bräysy, O., & Gendreau, M. (2005a). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science, 39*(1), 104–118.

Bräysy, O., & Gendreau, M. (2005b). Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation Science, 39*(1), 119–139.

Campbell, A. M., & Savelsberg, M. (2004). Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science, 38*(3), 369–378.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*(4), 568–581.

Dell'Amico, M., Monaci, M., Pagani, C., & Vigo, D. (2006). *Heuristic approaches for the fleet size and mix vehicle routing problem with time windows*. Working paper, University of Modena, Italy.

Dondo, R., & Cerdá, J. (2007). A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research, 176*(3), 1478–1507.

Dueck, G., & Scheurer, T. (1990). Threshold accepting: A general purpose optimization algorithm. *Journal of Computational Physics, 90*(1), 161–175.

Gehring, H., & Homberger, J. A. (1999). Parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. In: K. Miettinen, M. Mäkelä & J. Toivanen (Eds.), *Proceedings of EUROGEN99* (pp. 57–64). Jyväskylä, University of Jyväskylä.

Kytöjoki, J., Nuortio, T., Bräysy, O., & Gendreau, M. (2007). An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Computers & Operations Research, 34*(9), 2743–2757.

Li, F., Golden, B., & Wasil, E. (2007). A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. *Computers & Operations Research, 34*(9), 2734–2742.

Liu, F.-H., & Shen, S.-Y. (1999). The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research Society, 50*(7), 721–732.

Or, I. (1976). *Travelling salesman-type combinatorial problems and their relation to the logistics of regional blood banking*. Ph.D. thesis, Northwestern University, Evanston, IL, USA.

Paraskevopoulos, D. C., Repoussis, P. P., Tarantilis, C .D., Ioannou, G., & Prastacos, G. P. (2007). A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. *Journal of Heuristics* [Online publication].

Simchi-Levi, D., Kaminsky, P., & Simchi-Levi, E. (2008). *Designing and managing the supply chain concepts strategies and case studies*. New York: McGraw-Hill.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research, 35*(2), 254–265.

Taillard, E., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J. Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science, 31*(2), 170–186.

Toth, P., & Vigo, D. (Eds.). (2001). *The vehicle routing problem*. *SIAM monographs on discrete mathematics and applications*. Philadelphia, US: Society for Industrial and Applied Mathematics.

Voudouris, C., & Tsang, E. (1998). Guided local search. *European Journal of Operations Research, 113*(2), 469–499.

Pasi P. Porkka

Modeling Time Capacitated Resource Allocation In Services Allowing For Split Tasks

Vehicle Routing Problem Approach

Department of Business Technology

January

2010

AALTO UNIVERSITY

SCHOOL OF ECONOMICS

UNPUBLISHED WORKING PAPER

Abstract

In traditional production, production resources are typically machines and workers with fixed locations. Tasks are brought to these locations as raw materials or components in well-planned production batches. Because a production resource with fixed location is often the bottleneck that limits company turnover, a typical objective in production scheduling is the maximization of capacity utilization of that bottleneck. Also services, such as hairdresser or a dentist, can have fixed locations where customers come to be served. Often all tasks go through the same production line or service and the effect of routing within a production facility can be ignored.

Besides resources with fixed locations, there are mobile resources to be moved between tasks with fixed locations. Examples of tasks of this kind can be services, such as delivery, cleaning or maintenance, or production such as the construction of a building. In the moving and allocation of production or service resources, time is an important constraint. The time of mobile resources consists of moving between tasks, working on tasks, and slack. Working time in tasks often being constant, the traveling time related routing becomes critical in the minimization of resource consumption.

In current mathematical modeling literature movements and task performance are typically modeled as costs only. In addition, for easier solution, problems are often reduced from routing AND scheduling problems into routing OR scheduling problems. This essay studies a resource allocation problem where traveling times between tasks and times needed to perform tasks are explicitly modeled. Traveling time between tasks is modeled as parameters and routes are selected by using binary variables. Working time and slack of resources are modeled as continuous variables. For that kind of modeling, there is a clear gap in literature.

The focus of this essay is on the savings effect of allowing more than one resource to work on a task instead of one. The resource time allocated for tasks is modeled by continuous variables which let the model decide how the time needed to perform tasks is allocated between different resources. Flexible splitting logic assigns each task to one, two or even more resources.

There are three main objectives in this essay. First, a reader is given an understanding about the capacity saving effects of task splitting by describing how delivery splitting affects costs in vehicle routing. Second, it is shown that, by allowing the time capacitated task splitting between resources, up to 50% capacity time savings can be reached compared with a model which does not allow task splitting. Third, a Mixed Integer Linear Programming (MILP) problem model on Time Capacitated Resource Allocation Problem with Split Tasks (TCRAPST) is presented and discussed. Applicability of the TCRAPST in real-world situations is also discussed.

Key Words: service resource allocation, set-up time, time capacitated planning, scheduling, services, MILP

List of Contents

List of Figures

List of Tables

1. Introduction

In traditional production at a factory, a production schedule feeds tasks to machines with fixed locations in well-planned production batches. In many services, such as hair cutting, mobile customers come to production resources with fixed locations. In these cases, the production planning problem is to allocate mobile input, such as raw material or customers, to be processed by immobile production capacity. Products, like routes or houses, and services, such as home health care, facility cleaning, waste collection and machine leasing, can also be created by mobile people and equipment whose routing between production sites and tasks with fixed locations, but different durations, has to be considered simultaneously with batching and scheduling. In the most complicated case both production resources and tasks are mobile and their location, routing and scheduling has to be matched and optimized. Production capacity being the limiting factor of throughput, maximizing the utilization of capacity by efficient scheduling becomes top priority. Demand being fixed, minimizing the capacity needed is a typical objective. This essay describes and models a capacity requirement minimization problem with mobile resource time allocated for performing tasks having fixed locations and different durations. The total resource time for working in tasks being predetermined, the objective is achieved by minimizing the total traveling time simultaneously allowing flexible time based task splitting among resources.

The time capacitated allocation of mobile people and equipment is a general problem not only in production or services but also in our every day life. In flexible production, machines typically have overcapacity and the bottleneck is the working time of multi-skilled work force capable of running many different machines. In many services, capacity time consists of moving between tasks as well as preparation for and working on tasks, after-task-steps before leaving tasks, set-downs, and possibly some idle time. In our personal life, we have numerous possible time consuming activities to select from. Besides allocation and scheduling of our own limited time, we often have to consider shared time capacitated equipment such as a car used by different family members.

In capacity planning, time capacitated modeling is often equivocated or unrealistically simplified. The production and/or set-ups of products or services are typically modeled as costs only. Their time capacity consumption is deliberately ignored. In addition, problems are often reduced from routing and scheduling problems into routing or scheduling problems by fixing either of the components already in the problem preprocessing stage.

This essay studies a Time Capacitated Resource Allocation Problem with Split Tasks (TCRAPST) where moving time and productive time are explicitly modeled in constraints and variables. Tasks can be completed by one or more resources by modeling the working time in tasks as a continuous variable that allows the splitting of work in tasks between several resources. This kind of flexible time capacitated task splitting exhibits substantial capacity time savings potential, and therefore increased feasibility, compared with a modeling approach where each task is allocated to one resource only.

The TCRAPST with its savings effects resembles the Split Delivery Vehicle Routing Problem (SDVRP) introduced by Dror and Trudeau (1987) and surveyed by Archetti and Speranza (2008). The SDVRP, however, typically minimizes the distance traveled instead of capacity time needed. Therefore, the literature research later in this essay concentrating on the SDVRP shows a clear research gap in modeling time capacitated task splitting in vehicle routing.

This essay has three objectives. First, examples with and without task splitting are presented to demonstrate and to better understand the capacity time saving effects. Second, simple examples are used to prove up to 50% capacity time savings potential of task splitting compared with a situation when task splitting is not allowed. Third, the TCRAPST is modeled as a Mixed Integer Linear Programming (MILP) model.

This essay is structured as follows. Chapter 2 presents a simple example of the task splitting related capacity time savings often critical to the feasibility of a time capacitated resource allocation problem. Chapter 3 first presents a literature research on the benefits of split deliveries in vehicle routing. It also shows that, in vehicle routing with time considerations, time capacitated splits have not been applied on uploading at clients' facilities. Finally, the modeling approach of the TCRAPST is compared with a Vehicle Routing Problem. Chapter 4 shows that even up to 50% capacity time savings are, in theory, possible in time capacitated planning when time capacitated task splitting between different resources is allowed. In Chapter 5, the TCRAPST is modeled as a Mixed Integer Linear Program. Chapter 6 describes a TCRAPST solution by example. Chapter 7 discusses the potential business applications of time capacitated resource allocation with splits. Chapter 8 concludes this essay.

2. Why Time Capacitated Task Splitting?

Working and traveling consume capacity which can be measured as time and wasted by inefficient scheduling. Because the total working time in tasks is often fixed, the savings from more efficient scheduling comes from the better routing of working resources and better allocation of working capacity between tasks. Sometimes the feasibility of a problem depends on whether tasks can be worked on by more than one resource. The savings and feasibility effects of time capacitated modeling with this task splitting can be illustrated by example.

Let's consider an example where there are three tasks that should be completed within an 8 hour long planning period. Working capacities (C_p) of workers [$p \in (1,\ldots, P)$] and capacity requirements (R_i) for the tasks [$i \in (1,\ldots,I)$, $j \in (1,\ldots, I)$, $i \neq j$] are expressed in hours. There are three tasks $I = \{A, B, C\}$ with traveling times ($D_{i,j}$) between tasks being $D_{A,B} = D_{B,A} = 0{,}25$h, $D_{B,C} = D_{C,B} = 0{,}5$h, and $D_{A,C} = D_{C,A} = 0{,}6$h. Capacity requirements in tasks are $R_A = 13$h, $R_B = 20$h, and $R_C = 27$h. Working capacity (C_p) is 8 hours for each worker. Figure 1 illustrates the initial stage in the example. Diameters of circles demonstrate the capacity requirements of tasks. Arrow labels mark traveling times between tasks.

An initial solution for the problem can be found by dividing task requirements by worker capacities ($R_i / C_p = R_i/8$) and then rounding up to the next integer, which indicates the initial capacity allocation for each task:

R_A/C = 13/8 rounds up to 2 workers and 2*8 = 16 capacity hours

R_B/C = 20/8 rounds up to 3 workers and 3*8 = 24 capacity hours

R_C/C = 27/8 rounds up to 4 workers and 4*8 = 32 capacity hours

This initial feasible solution satisfies customer requirements but includes a considerable amount of slack (h_p) indicated as unallocated working hours of workers. On the other hand, none of the allocated workers needs to travel between tasks during his work shift.

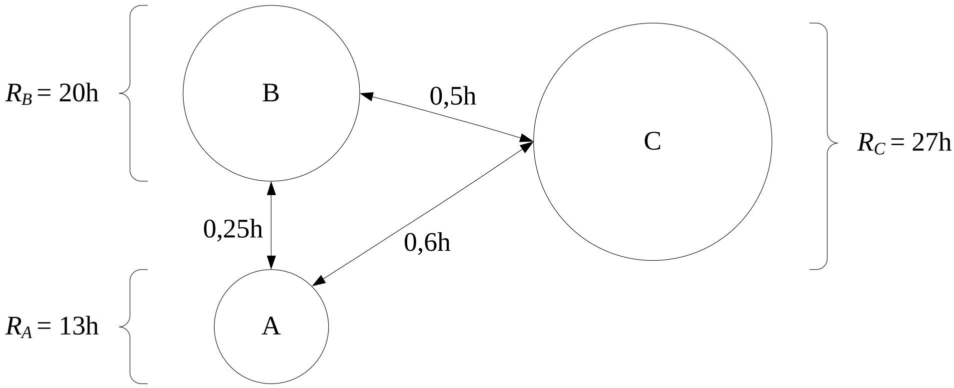

Figure 1. Initial stage of time capacitated split task example with three tasks

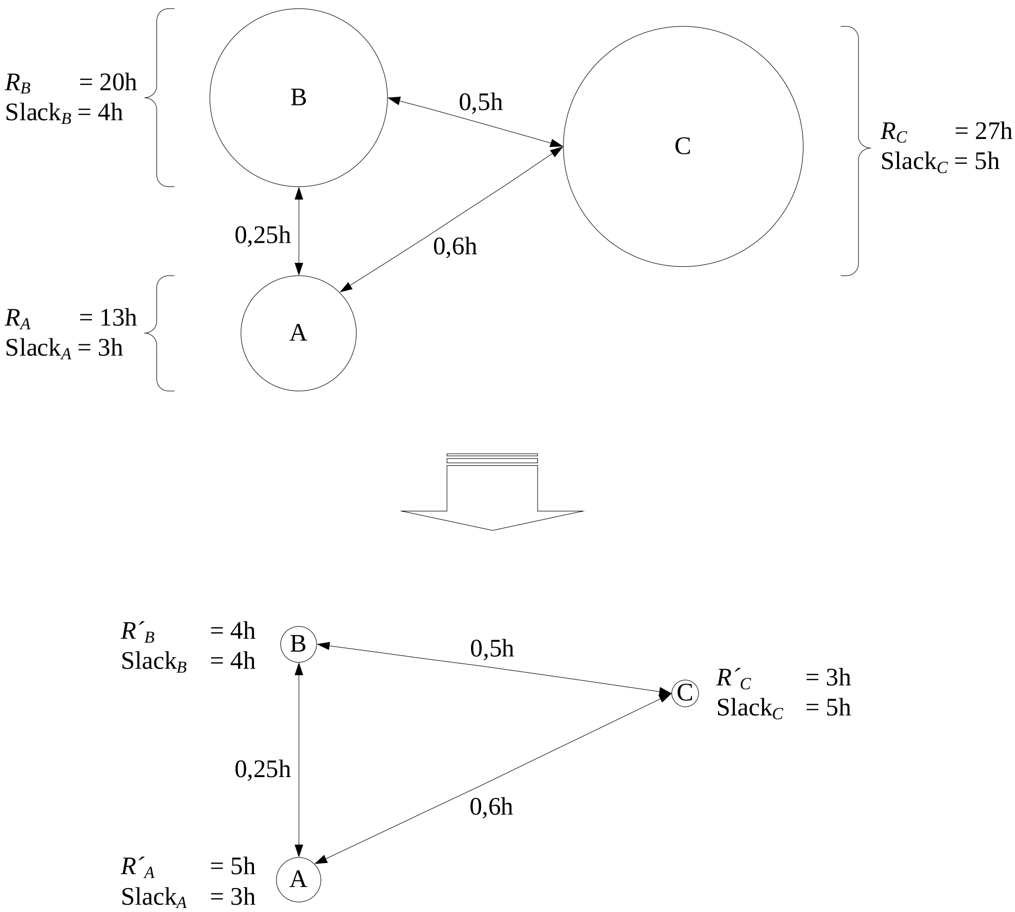

Figure 2. Reduction of the problem

We can also calculate the amount of slack per task as a difference between working capacity requirements in tasks and allocated worker capacities:

$$\text{Slack}_A + \text{Slack}_B + \text{Slack}_C = (16 - 13) + (24 - 20) + (32 - 27) = 3 + 4 + 5 = 12$$

As we now can see, working capacity is required to perform work for just (13 + 20 + 27)/8 = 59,5/8 = 7,5 working days, but our initial solution allocates 9 workers to work for 9 working days!

The initial solution can, however, be improved by allowing workers to be reassigned during a planning period. Before the reallocation, we can simplify the problem by assuming for each task that only workers with slack capacity can be reallocated. In that way, we can reduce the problem by assuming the maximum of only one worker with slack for each task and by removing the same number of required working hours in each task as there are capacity hours allocated for full-time workers. Workers without slack are thus assumed to use their whole working capacity working for one task only. As full-time workers are removed, there remains a maximum of one worker with slack capacity in each task. By making these assumptions, we can reduce the problem and concentrate on reassigning only workers with slack capacity.

Figure 2 illustrates the original problem and the reduced problem where required working hours are $R'_A = 5\text{h}$, $R'_B = 4\text{h}$, and $R'_C = 3\text{h}$. Because the total slack ($\sum_{p=1}^{P} h_p = 3+4+5 > 8$) of workers exceeds the capacity of one worker, we can try to reduce the number of needed workers by reallocating workers during a work shift.

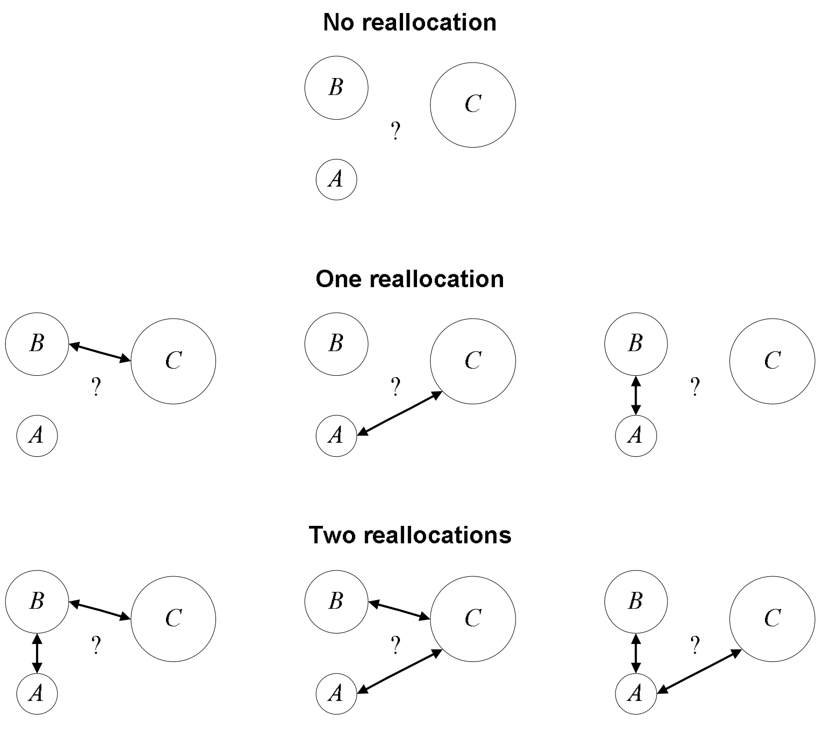

Figure 3. Reallocation combinations of two workers with slack capacity

Figure 3 illustrates our options to reallocate workers in our example. We either do not reallocate at all or we can reallocate one or two workers. Reallocating all three workers would not make sense. If we reallocate one worker, that worker is supposed to complete two tasks. If we reallocate two workers, we have to decide how to divide work between workers in one of the tasks.

When we reallocate workers, we should take into account time needed to travel between tasks. If we do not include this set-up time into our calculations, we can get solutions that can not be applied in practice.

Table 1 and Table 2 demonstrate the importance of set-up time. In both tables, parentheses are used to indicate work performed by one worker. In Table 1 traveling times are not included, but in Table 2 they are.

In Table 1, where traveling time is not included, we see two feasible solutions that decrease the number of workers by one. In the first row the solution is not feasible because one worker should work for 9 hours which is more than his capacity. In second row the solution is feasible as one of the workers distributes his whole working time between tasks *A* and *C*. The other worker working in task *B* incurs 4 hours of slack. In third row the solution is feasible as one of the workers distributes his working time between tasks *B* and *C* incurring 1 hour of slack. The other worker working in task *A* incurs 3 hours of slack.

Table 1. Worker reallocation when traveling times are not included

Work Assignments	Work	Traveling Time	Used Capacity	Feasibility
($A \leftrightarrow B$); (C)	**(5,0 + 4,0); (3,0)**	**-**	**~~(9,0)~~; (3,0)**	**Infeasible**
($A \leftrightarrow C$); (B)	**(5,0 + 3,0); (4,0)**	**-**	**(8,0); (4,0)**	**Feasible**
($B \leftrightarrow C$); (A)	**(4,0 + 3,0); (5,0)**	**-**	**(7,0); (5,0)**	**Feasible**

In Table 2, where traveling time (or set-up time) is included, we get only one feasible solution. The solution on the first row would be infeasible without the inclusion of traveling time, but now also the solution on the second row becomes infeasible because working time added with traveling time is more than 8 hours for one of the workers. Only the solution on the third row is feasible because neither of the workers work or travel over their capacity and both of the workers have slack even after the transportation of one worker between two tasks.

Table 2. Worker reallocation with traveling times

Work Assignments	Work	Traveling Time	Used Capacity	Feasibility
($A \leftrightarrow B$); (C)	**(5,0 + 4,0); (3,0)**	**0,25**	**~~(9,25)~~; (3,0)**	**Infeasible**
($A \leftrightarrow C$); (B)	**(5,0 + 3,0); (4,0)**	**0,60**	**~~(8,60)~~; (4,0)**	**Infeasible**
($B \leftrightarrow C$); (A)	**(4,0 + 3,0); (5,0)**	**0,50**	**(7,50); (5,0)**	**Feasible**

Let's now modify the example by changing the capacity requirement R_C = 27h to R_C = 27,75h and the reduced capacity requirement R'_C = 3h to R''_C = 3,75h. The new reduced example is illustrated in Figure 4. Because the total of reduced required work time in *B* and *C* plus traveling time between those tasks now exceeds the capacity of one worker ($R'_B + R''_C + D_{B,C} = 8{,}25 > 8$), we can no more find a feasible 8 worker solution for the unreduced problem by moving just one worker. To find a feasible solution, we have to move two workers instead, as in the last three possible combinations in Figure 3.

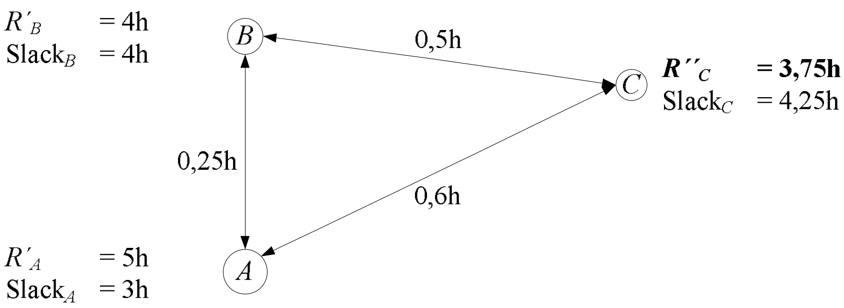

Figure 4. Reduced problem with working time requirement R''_C = 3,75h

We still need to find a criterion to select between alternative feasible solutions. One criterion could be to minimize the total traveling time of the transferred workers because in this way we can identify the maximum amount of slack that could be allocated for other tasks.

In Table 3 capacity consumptions of different reallocations are calculated. The optimal solution (*) that maximizes slack appears to be on the first row where the task *B* is shared by two reallocated workers. In each of the solutions in Table 3, there is still slack to be allocated to perform other tasks.

Table 3. Worker reallocation alternatives with two reallocations

Assignments	Work	Traveling Time	Used Capacity	Slack
(A ↔ B); (B ↔ C)	**5,0 + 4,0 + 3,5 = 12,5**	**0,25 + 0,50 = 0,75**	**13,25**	**2,75***
(B ↔ A); (A ↔ C)	**4,0 + 5,0 + 3,5 = 12,5**	**0,25 + 0,60 = 0,85**	**13,35**	**2,65**
(A ↔ C); (C ↔ B)	**5,0 + 3,5 + 4,0 = 12,5**	**0,60 + 0,50 = 1,10**	**13,60**	**2,40**

This little example demonstrated that savings can be reached by the reallocation of resources between tasks and by splitting work in tasks among more than one resource. In this case, finding an optimal solution was easy. However, when the number of tasks and workers increases, finding an optimal solution without mathematical modeling gets impossible.

3. Vehicle Routing Problems

The time capacitated allocation of workers to tasks is a routing problem and the modeling approach taken in this essay is very similar to the well-known Split Delivery Vehicle Routing Problem (SDVRP). This chapter describes the Vehicle Routing Problem (VRP) and gives a literature research on the SDVRP to show a research gap.

The VRP concerns the distribution of goods between depots and final users. Customers and typically one depot form a network usually modeled as a graph which can be either directed or non-directed. Typically transportation capacity or route length is limited leading to a situation when all customers can not be served by one route and one vehicle only. Other constraints can include periods of the day (time windows) during which customers have to be served, unloading or loading times, vehicle type, different priorities, and penalties associated with partial or total lack of service associated with customers. Routes can include deliveries, pick-ups or both. The objective is to minimize transportation costs that consist of the number of vehicles needed and actual traveling costs typically consisting of the total distance traveled.

If every client must be serviced by exactly one vehicle, the problem is known as the Capacitated Vehicle Routing Problem (CVRP) which has been the focus of intensive research in the last 25 years. Toth and Vigo (2001) wrote a comprehensive book on Vehicle Routing Problem models and algorithms to solve them. A VRP problem allowing a client to be served using more than one vehicle is a SDVRP. The SDVRP is a relaxation of the classical VRP, but it still remains NP-hard.

The SDVRP is quite similar to the problem of allocation of moving resources to services. In the SDVRP a customer requiring products can be served by one or more vehicles whereas in service resource allocation a task can be completed by one or more service resources. In both cases routing and split decisions are required and the objective is to minimize resource costs. Considering vehicle routing problems, the following literature research clearly shows that by allowing split deliveries savings can be reached in both the total distance traveled and in the number of vehicles to be used.

The time spent at customer’s site, as products are picked up or delivered, could be considered as a constraint on vehicle routing too. In that case, the delivery splits could be based on the time that it takes to pick up and to unload products to and from a vehicle.

Sometimes both the delivery amount and service time may be constraints. For example, pumping oil from or into big tankers can take a long time and may not always be completed in one dock because the dock may need to be freed for another tanker loading or unloading another type of oil. One tanker may be able to satisfy loading or unloading requirements only partially before it has to leave. Loading and unloading of oil in tankers takes a long time and sometimes time at a dock can become a constraint.

The splitting of deliveries is not exactly the same as the splitting of time capacitated tasks, but they both have the same kind of savings potential. Therefore, understanding split deliveries in vehicle routing creates intuition of using split tasks in time capacitated service resource allocation. The next subchapter provides a literature research on the SDVRP to give that understanding. It also shows a clear research gap in time capacitated resource allocation with split tasks.

3.1. Literature Research on Split Delivery Vehicle Routing Problems

In this literature research the focus is on the SDVRP which is not covered by Toth and Vigo (2001) in their book on the VRP. The purpose is to better understand split deliveries and to find out whether time spent with clients has been used as the basis of splitting decision.

VRP problems typically assume that customers are served by one vehicle only. If customer’s requirement exceeds the capacity of a vehicle, the problem can be reduced by assigning routes that travel from depot to one customer with a full load and then directly back to the depot. Then, only the routing for customer demands exceeding an integer number of vehicle capacity is modeled and solved. After solving the reduced problem, one-customer-routes are added to the plan.

A more recent approach has been to allow the splitting of the deliveries or pick-ups of a customer between one or more vehicles. Dror and Trudeau (1987) introduced the concept of Split Delivery Routing. Dror and Trudeau (1989) presented a Split Delivery Vehicle Routing Problem (SDVRP), developed a solution scheme and demonstrated the potential for cost savings through split deliveries by using generated problems. Their solution scheme is a two-stage algorithm, where the first stage constructs a VRP solution using node interchanges, and the second stage improves the VRP solution by introducing and eliminating splits. Their algorithm found almost 14% cost reductions when costs for a set of randomly generated problem instances were solved as both a VRP and an SDVRP, with computational times under 30 minutes for the largest problems with 150 customers. In their approach split loads were selected by determining the cost savings found by removing a load from a route and servicing portions of the load on at least two other routes. The loads were divided based on the available vehicle capacity on other routes. They also show that no two routes can have more than one split in common, which greatly limits the number of splits that have to be analyzed.

Dror and Trudeau (1990) further elaborate their study and prove that two points whose demand is supplied by a number of routes do not have more than one route in common when the distance matrix satisfies the triangular inequality. They also show that the SDVRP is an $\mathcal{NP}$-hard problem. Dror et al. (1994) presented an integer formulation of the SDVRP and developed different classes of valid inequalities and used these constraints in a cutting plane algorithm to solve small instances with ten clients to optimality. They also developed an exact constraint relaxation Branch-and-Bound algorithm for the SDVRP.

Besides Dror and Trudeau, Brenniger-Göthe (1989) made early research on the SDVRP. She presented the SDVRP in her doctoral dissertation and applied it to a distribution planning case.

Federgruen and Simchi-Levi (1995) split the demand d of a client into d clients with unit demand and same location. Then they solve the problem with different heuristics whose performance is compared.

Sierksma and Tijssen (1998) apply split demands in determining a flight schedule for helicopters to off-shore platform locations for exchanging crew people employed on these platforms. They propose a Cluster-and-Route Heuristic algorithm for short term planning as well as an exact long-term planning LP model and solve it by column generation techniques. Exact solutions are found with up to 50 platforms. However, some rounding is needed to obtain an integer solution.

Important properties in their model are that the demanded crew exchanges of a platform need not be carried out by one helicopter and crew exchanges can not be fractional. So, the problem is a discrete split delivery routing problem. Two major constraints on their problem are the capacity of the vehicles and the range that helicopters can fly. By an appropriate splitting strategy, a range limited infeasible plan can be changed into feasible.

In their example in Figure 5, there are three platforms (P_1, P_2, and P_3) with demands (1, 18 and 1) and the capacity of a helicopter is 10. If the helicopter first flies to P_2 and exchanges 10 crew members (Airport – P_2(10) – Airport), then the second route consisting of exchanges at all three platforms (Airport – P_3(1) – P_2(8) – P_1(1) – Airport) may exceed the range making the solution infeasible. The other schedule with (Airport – P_2(9) – P_1(1) – Airport) and (Airport – P_2(9) – P_3(1) – Airport) has a shorter longest route that may be acceptable.

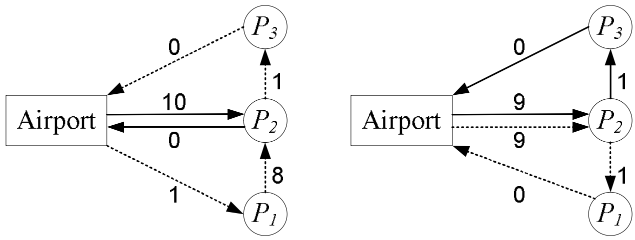

Figure 5. The limited range of the helicopters

Belenguer et al. (2000) define a solution of the SDVRP and show that the convex hull of the associated incidence vectors is a polyhedron, whose dimension depends on whether a vehicle visiting a client must service, or not, at least one unit of the client demand. They present a new family of valid inequalities and lower bounds that are used to get the optimal resolutions of some known instances with up to 50 clients.

Archetti et al. (2001) studied a split delivery problem with discrete capacities 2 and 3, possibly greater integer demands than 2 and 3, and general distances. They show that if some specific conditions on the distances are satisfied, the problem with a capacity of 2 is solvable in polynomial time. When the distances are symmetrical and satisfy the triangle inequality, this problem is reducible, by making direct trips to the depot, to a problem where each customer demand is strictly lower than 2. They also show that the problem with vehicle capacity $k > 3$ is $\mathcal{NP}$-hard. When the capacity is equal to 3 the problem is reducible only when the distances satisfy the sharpened triangle inequality with $\alpha = 2/3$.

Archetti et al. (2005) studied a Skip Delivery Problem (SDP) where a fleet of vehicles must deliver skips to a set of customers. In their problem, each vehicle has a maximum capacity of two skips. Tours have to start and end at a central depot, and the demand of each customer can be greater than the capacity of the vehicles. They show that the SDP is solvable in polynomial time, while its generalization to the case where all vehicles have a capacity greater than two, known as the SDVRP, is shown to be $\mathcal{NP}$-hard, even under restricted conditions on the costs. The demands in this SDP being integers, costs symmetrical and satisfying the triangle inequality, it can be shown that the SDP can be reduced in polynomial time into a problem of possibly smaller size, where each customer has unitary demand. This property allows a remarkable simplification of the problem.

Bompadre et al. (2006) presented improved lower bounds for the SDVRP and solved the problem using a quadratic iterated tour partitioning (QITP) heuristic. They solve the SDVRP by transforming it into a unit-demand problem, where each customer i with demand d_i is replaced by a clique of q_i customers with unit demand each and zero interdistance.

Archetti, Speranza and Hertz (2006) use a tabu search procedure for the SDVRP. They consider a k-SDVRP where a direct trip is a tour where a vehicle starts from the depot, goes directly to a customer, where it delivers k units, and then turns back directly to a depot. Given an instance I of the k-SDVRP, one can build a reduced instance, denoted I_R, by modifying the demand d_i of each customer to $d_i - k[d_i/k]$. A solution S_R for I_R can then be transformed into a solution s for I by adding $[d_i/k]$ direct trips for each customer i. Now, given an instance I of the k-SDVRP, consider the algorithm that first determines an optimal solution s_R^* for the reduced instance I_R and then builds the associated solution s^* for I.

Their tabu search algorithm overcomes the typical problem of the tabu search algorithms: the tuning of the parameters. Actually, only two parameters, the length of the tabu list and the maximum number of iterations the algorithm can run without improvement of the best solution found, have to be set. At each iteration, they obtain a neighbor solution by removing a customer from a set of routes where it is currently visited and inserting it either into a new route or into an existing route that has enough residual capacity. The insertion of a customer into a route is done by means of the cheapest insertion method. Their approach performs well against Dror and Trudeau (1989) and almost always provides better solutions on the tested instances even with very short computation times.

Lee et al. (2006) considered a multiple-vehicle routing problem with split pick-ups (mVRPSP). Their problem involves multiple suppliers, a single depot, and a fleet of identical capacity trucks responsible for delivering supplies from suppliers to the depot. The problem is basically the same as the SDVRP, but the solution approach is new. Lee et al. develop a deterministic dynamic (DP) program to solve their problem exactly and use an optimality-invariance condition to find formulations that lead to equivalent DP with finite state and action spaces. Solving these DP formulations is based on a shortest path search algorithm, which is conceptually simple and easy to implement.

Chen et al. (2007) reviewed the applications of the SDVRP and developed a heuristic that combines a mixed integer program and a record-to-record travel algorithm. Their heuristic generally performs much better than the tabu search heuristics from Archetti et al. (2006). Chen et al. (2007) also give an example of split pick-ups in commercial sanitation collection where clients often place their trash in large containers or bins. A building may have several bins which are lifted and their contents are emptied into the trash truck. Several trucks may be required to pick up all the trash at a particular office building. However, because a bin's load can not be split, each bin either has to be considered as a separate demand or the building is handled as a single demand. In the latter case, a discrete number of splitting options is allowed.

Mota et al. (2007) presented a scatter search base methodology constructed for the SDVRP. They do not refer to Chen et al. (2007), but consider in their tests the same set of instances used by Archetti et al. (2005). With some instances, they manage to get better solutions than Archetti et al. (2005) in reasonable computing time.

Jin et al. (2007) proposed a two-stage (TS) algorithm with valid inequalities (TSVI) to optimally solve the SDVRP. The first stage in the TSVI creates clusters covering all demand and establishes a lower bound. The second stage calculates the minimal distance traveled for each cluster by solving the corresponding traveling salesman problem (TSP). The sum of the minimal distance traveled over all clusters yields an upper bound. Their numerical experiments show that TSVI significantly outperforms other exact solution approaches provided in the literature for the SDVRP. The paper does not refer to the heuristics in Chen et al. (2007).

Mizrak Özfirat and Özkarahan (2007) formulated the Heterogenous fleet VRP (HVRP) with and without split deliveries for fresh food distribution of a retail chain store. They proposed algorithm that decomposes the main problem into subproblems and simultaneously allocates vehicles to a number of $\mathcal{NP}$-complete subproblems. Then they employed integer programming to solve subproblems. In their case company, the improvement achieved by split delivery strategy compared to non-split delivery was only 0.1%.

Tavakkoli-Moghaddam et al. (2007) present a mixed integer linear model of a CVRP with split services and heterogenous fleet. Then they solve it by using a simulated annealing method.

Campos et al. (2007) developed a scatter search method based algorithm that produces feasible solutions using the minimum number of vehicles. Compared with the algorithm from

Archetti et al. (2006), their algorithm performs well when demands are well below half the capacity, but not so good when demands are over half the capacity.

Jin et al. (2008) developed a column generation approach to the SDVRP with large demand. Their columns include route and delivery amount information. Pricing sub-problems are solved by a limited-search-with-bound algorithm. Feasible solutions are obtained iteratively by fixing one route once. Their numerical experiments show better solutions than in the literature.

Jin et al. (2008) generated an approach resembling the work of Sierksma and Tijssen (1998) for helicopter routing. Their approach, however, differs in three ways. Firstly, they require the solutions to use a fixed number of vehicles but do not have a restriction on the longest distance of each trip. Secondly, their decision variables in the master problem are defined as binary variables rather than general integer variables in order to improve the lower bound. Thirdly, their algorithm to obtain the feasible solution fixes one route with the largest product of variable value and its total delivery guaranteeing feasibility over iterations. They conduct numerical experiments on the instances published in Belenguer et al. (2000) and show that the column generation approach provides good results for instances with large average demand.

Mitra (2005) formulate a Vehicle Routing Problem with Split Deliveries and Pickups (VRPSPDP) as a MILP and develop a route construction heuristic. In their model, the linehaul and backhaul customers are allowed to be the same which leads to simultaneous delivery and pickup at a customer location. Their model either did not include any restriction on the quantity demanded at (to be returned from) a customer location. A customer may be visited by more than one truck and more than once by the same truck. In Mitra (2008) they use the same problem sets given in Mitra (2005), give an alternative MILP formulation and develop a parallel clustering technique to arrive at an initial solution to the problem.

Nowak et al. (2008) applied split loads for Pickup and Delivery Problem (PDP) calling their problem a Pickup and Delivery Problem with Split Loads (PDPSL). They solved the problem by using a heuristic to quantify the benefits of using split load for some large-scale random instances. In a real-world trucking industry problem, these benefits of the heuristic were, however, reduced.

According to Nowak et al. (2008) the PDPSL is a more complex problem than the SDVRP primarily because the available capacity of the vehicle changes each time a load is picked up or delivered for the PDPSL, without the vehicle ever returning to a depot. With the SDVRP load planning is done with the same fixed capacity prior to a vehicle leaving the depot. With the PDPSL available capacity during a route changes depending on deliveries and pickups which makes it difficult to determine where to insert a split load. Some of the SDVRP techniques are, however, still applicable to PDPSL. For a PDP literature review see Nowak et al. (2008).

In 3PL case study Nowak et al. (2008) used an additional monetary cost associated to the time needed for vehicle loading or unloading at a facility. They didn’t, however, use actual time as a resource.

Archetti, Speranza and Savelsbergh (2008) use an optimization based heuristic to solve the SDVRP. As the solution approach by Chen et al. (Chen, Golden et al. 2007), their solution approach also integrates heuristic search with optimization by using an integer program to explore promising parts of the search space identified by a tabu search heuristic. Computational results show that the method improves the solution of the tabu search in all but one instance of a large test set.

A comprehensive description of applications and solution approaches to the SDVRP can be found in Chen et al. (2007). Recently, the basic SDVRP has been extended to include also pick-ups (see Lee et al. (2006), Nowak et al. (2008)) and time windows (see Feillet et al. (2003), Ho & Haugland (2004), Gendreau et al. (2008)).

3.2. Split Delivery Vehicle Routing Problem with Time Considerations

In models reviewed in the last subsection time is ignored as a resource that is consumed when service is provided. Because in the TCRAPTS both movements between tasks as well as work in tasks are modeled as time that they consume, it is important to know how time has been taken into account in literature.

Gendreau et al. (2008) determine the Vehicle Routing Problem with Time Windows (VRPTW) as a problem consisting of determining a least cost set of vehicle routes such that every route starts and ends at the depot and such that every customer is served exactly once by one vehicle, the vehicle capacity being respected and the service of customers beginning inside their time windows. A vehicle might arrive at a customer location before the beginning of its time window and wait. A variant problem of the VRPTW is when the requirement that each customer is served exactly once is relaxed. The demand d_i of a customer can then be divided arbitrarily among the vehicles visiting him. This problem is known as the Split Delivery Vehicle Problem with Time Windows (SDVRPTW).

Frizzell and Griffin (1992) consider the problem on a grid network and in Frizzell and Griffin (1995) they add time window constraints to the problem creating the Split Delivery Vehicle Routing and Scheduling Problem with Time Windows (SDVRSPTW). In both cases, they propose heuristics to solve the problems. Their proposed heuristics are especially tailored for the problems and this kind of grid network structure. They also include an explicit splitting cost which is typically ignored in other split delivery papers.

Frizzell and Griffin (1992) consider the problem on a grid network and in (1995) they add time window constraints to the problem creating the Split Delivery Vehicle Routing and Scheduling Problem with Time Windows (SDVRSPTW). In both cases, they propose heuristics to solve the problems. They also define delivery time as a non-linear function of the delivered amount exhibiting the economies of scale. The extra costs of allowing any split deliveries are also considered. The time required to unload a vehicle is considered within the delivery time, while the time to load a vehicle is arbitrarily chosen to be 30 min.

Frizzell and Griffin (1995) use five performance measures: drive time, which is the total time a vehicle is utilized (this includes travel time, loading and unloading); route number, which is the number of vehicles used; split deliveries, which is the number of customers whose demand is split; waiting time, which is the time a vehicle spends at customer location while not in the process of making a delivery; and lag time, which is the remaining time in the day which is neither drive time nor waiting time.

Mullaseril et al. (1997) develop a heuristic algorithm for a cattle feeding problem in a ranch. Cattle are kept in large pens that are connected by a road network. Six trucks deliver feed to the pens within a specific time window each day. Because of feed weighing and loading inaccuracies, the last pen on a route may not receive its full load and would need to have the rest of its load delivered by a second truck on a different route. They model the problem as a Capacitated Arc Routing Problem with time window constraints, where the demand of an arc may be split. The problem is a split delivery capacitated rural postman problem with time windows on arcs. Their heuristic is similar to the one proposed by Dror and Trudeau (1989) and (1990). The computational tests show that allowing split deliveries significantly reduce the total distance traveled by the fleet in most of the cases.

Mullaseril et al. (1997) included time windows in their cattle feeding model. However, their article does not clearly indicate whether the reduced discharging time in split deliveries was taken into account.

Feillet et al. (2003) present a Branch and Price approach for solving the SDVRPTW without imposing restrictions on the split delivery options. In their model, a time window and a service time, not dependent on the quantity delivered, is defined for every customer. The problem is presented as a set covering formulation without any assumption on the way in which demands are split. In other words, the proportion of demand delivered by each vehicle by each customer is a continuous linear variable. They solve problems of moderate size to optimality by using column generation and valid inequalities with an adapted classical branching scheme.

Song et al. (2002) considered the distribution of newspapers from printing plants to agents. They included in their model the possibility to split deliveries for agents located close to printing plants so that the split delivery spreads out agents' work of inserting supplements and allows home delivery to start earlier. Agents far from the printing plants and agents with small demands are not considered for split deliveries.

Song et al. (2002) used a two-phase solution procedure. In Phase I, they allocated agents to plants by solving a 0–1 integer programming problem. In Phase II, the authors determined the split deliveries, generated the vehicle routes using a modified savings rule and a weighted savings rule, and scheduled the vehicles for dispatch. Their method brought an average of 15% savings in delivery costs and reduced the delay time by 40% in comparison to the method used in their case company.

Song et al. (2002) use in their model both production time and traveling time. However, they do not consider the time spend at client's location when uploading.

Ho and Haugland (2004) use tabu search heuristic for the Vehicle Routing Problem with Time Windows and Split Deliveries (VRPTWSD). They apply four common move operators while simultaneously generating split loads. The split loads are created based on the amount of available capacity on a route when a load is to be placed on the route. This heuristic was used to solve the Solomon (1987) test problems with 100 customers in less than 35 minutes. The model used by Ho and Haugland (2004) includes traveling times but does not include service time for uploading by clients.

Gendreau et al. (2008) present a column generation approach for solving the Vehicle Routing Problem with Time Windows and Split Deliveries (SDVRPTW). Typically, only a limited number of splitting possibilities has been exploited in the modeling of the SDVRP problem and in most of the formulations the proportion of the demand of a customer served by a vehicle has been determined to be an integer number. Gendreau et al. (2008) propose a new set covering formulation for the SDVRPTW and describe a complete algorithm that solves instances of moderate size to optimality, without making any assumption of the fashion way in which demands are split. However, considering the service time, they make an assumption that the service time of customers does not depend upon the quantity delivered. This assumption can be done when "paper work" associated with each delivery is done in parallel to the unloading operation and dominates the unloading time.

3.3. Time Based Allocation of Service Resources with Split Tasks in Literature

Study of the current literature in Vehicle Routing Problems with Split Delivery indicates that time has not yet been considered as a resource or capacity in the similar way as it has been considered in TCRAPTS formulated later in this essay. Time has been considered as traveling time, but the time spent at client's location, except in Frizzell and Giffin (1992, 1995), is always fixed and does not depend on the amount of uploaded products. If working time is considered as a resource, the current SDVRP models do not model it realistically enough for resource allocation planning with time considerations and split tasks.

In most of the SDVRP models only some predefined discrete proportions of splits are allowed (For example, Sierksma and Tijssen (1998), Archetti et al. (2001), Archetti et al. (2005), Bompadre et al. (2006), and Tavakkoli-Moghaddam et al. (2007).). Some model documentations (Mullaseril et al. (1997) and Belenguer et al. (2000)) leave it unclear whether the splitting variable is discrete or continuous. The only articles using continuous or practically continuous split variables were from Feillet et al. (2003), Mitra (2005), Mitra (2008) and Gendreau et al. (2008).

In literature, modeling services with capacity consuming set ups and flexible works sharing has not yet received much attention in operations research community. According to author's knowledge, the most similar research to the TCRAPST outside the VRP literature is the work by Eveborn et al. (2006) where they plan home care staff reallocation between tasks within work shifts. The efficiency of the plan is judged by the amount of travel time it requires and how well it has succeeded in allocating all visits to staff members. The quality is judged by how well continuity is kept with staff member visits to each client. The model of Eveborn et al. (2006) is for planning a sequence of task allocations several days before the tasks are to be completed. Still, they do not include time capacitated splits of tasks between workers.

3.4. Benefits of Split Deliveries in the VRP

A typical savings of allowing splits could be a problem with 20 customers where a VPR solution suggests 8 routes but a solution with splits reduces the number of routes to 7 by including a split delivery for 2 customers. Figure 6 illustrates the difference between the VPR and the SDVRP. Nowak (2005), Lee et al. (2006), and Nowak et al. (2008) have applied the idea of splits for pick-ups too. On the other hand, in some cases it can also be beneficial to remove a split, and use more routes instead.

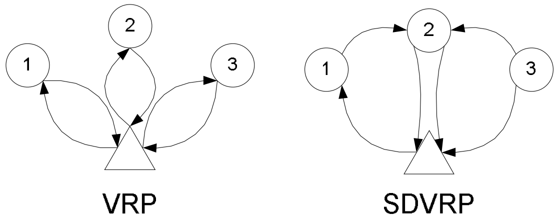

Figure 6. The VRP and the SDVRP solution

Splitting deliveries in the SDVRP can save in routing costs in comparison to the VRP that does not allow split deliveries. Archetti, Savelsbergh and Speranza (2006) show that the reduction in delivery costs from allowing split deliveries is at most 50%. In their analysis, they use discrete demand. They also demonstrated a strong relationship between the reduction in delivery costs and the reduction in number of delivery routes resulting from allowing split deliveries.

According to Archetti, Savelsbergh and Speranza (2008), the major benefit of allowing split deliveries is a possible reduction in number of vehicles. A substantial reduction in distance traveled when allowing split deliveries is typically the result of a reduction in the number of delivery routes and thus vehicles needed.

In their overview of the SDVRP Archetti and Speranza (2006) state that the benefits from allowing split deliveries mainly depend on the relation between mean demand and vehicle capacity and on demand variance; there does not appear to be a dependence on customer locations.

Arghetti, Savelsbergh and Speranze (2008) show that

- When demands are large relative to the vehicle capacity, then there is little advantage to splitting deliveries;
- When demands are small relative to the vehicle capacity, then there is little advantage to splitting deliveries;
- When demands are little over half the vehicle capacity, then there may be substantial advantages to splitting deliveries.

In Pickup and Delivery Problem with Split Loads (PDPSL) Nowak et al. (2008) found the benefit of split loads being most closely tied to three characteristics: load size, cost associated with a pickup or delivery, and the frequency with which loads have origins or destinations in common. Most benefit can occur with load sizes just above one half of vehicle capacity when loads with size close to vehicle capacity or below 10% of capacity showed almost no benefit. The benefit of load splitting was also often reduced by additional stops at origins or destinations, which increased the cost of making a pickup or delivery. However, splitting a load does not necessarily result in the addition of stops to a route if several loads share a common origin or destination. Sometimes cost savings may be reduced but other benefit may come from reduction in the number of routes. In PDPSL, the most benefit with split loads is found through route length reduction. Authors conclude that the benefit of split loads depends on conditions and rules in their 3PL case company, where almost all the cost savings were eliminated even, when there was a reduction in the number of routes used for service. Because of limited real benefits and because split deliveries can also lead to higher customer inconvenience due to more complex administration and accounting, companies need to carefully evaluate trade-offs of including split deliveries in their schedules.

3.5. Justification of Vehicle Routing Approach

Vehicle routing modeling can be associated with moving resources and the sharing of capacity within tasks. The vehicle routing problems typically include delivery requirements that can be served by one or more vehicles. Considering vehicle routing problems, the literature clearly shows that the splitting of deliveries can bring up to 50% savings in vehicle routing. Even though the splitting of deliveries is not the same as the splitting of time, understanding split deliveries in vehicle routing helps to see the potential of savings that could also be reached when splitting resource time between tasks. In addition, the time spent at customer’s site as products are picked or delivered could be considered as a constraint on vehicle routing too. In that case, the delivery splits could be based on the time that it takes to pick up and to upload products to and from a vehicle. Besides vehicle routing we could also observe resource capacity allocation in many other allocations including human resource or robot allocation and routing for different tasks. Comparison of time capacitated splits to other approaches than vehicle routing is left for further research.

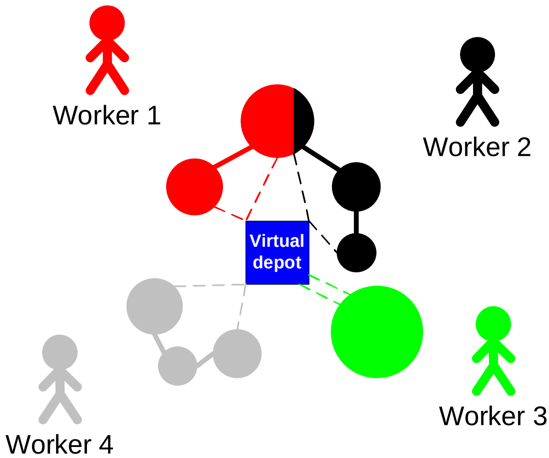

Figure 7. Illustration of paths generated by the TCRAPST

Figure 7 illustrates a modeling approach used in this essay. The approach is similar to vehicle routing modeling where routes start and end in a depot. We can, namely, use paths that resources travel and the paths start and end with a virtual depot with zero traveling times for entering and leaving the depot. User of the solution does not see the virtual depot. Instead, for a user the solution is shown as paths that start from a task, go via tasks and end in a task. Paths are not directed which means that either end of a path can be a starting point. A path can also consist of one task only. If a real-world application requires a base where resources are located when they are not working, the base (or depot) and routes leaving and entering it can be made visible and given actual transportation times.

Figure 7 illustrates the paths of four workers. One task is shared by Worker 1 and Worker 2. One task of Worker 3 is big enough to consume the whole capacity of that worker. Worker 4 has three tasks and does not have any shared tasks with other workers.

4. Savings from Flexible Task Splitting in the TCRAPST

For their PDPSL Nowak et al. (2008) define a split route as follows:

> *If load is partitioned into more than the minimum number of divisions for full service, it is considered to be split. For example, if a load is 3.6 truckloads and vehicle capacity is 1, a minimum of four trips are required to fully service the load. If the load is serviced in five or more trips, then it is considered to be split.*

The TCRAPST either does not limit the number of resources serving in a task. However, if there is more than one resource serving in a task, the task is considered to be split.

In the same way, as load splitting in the SDVRP can result in considerable savings, we can also show the potential of savings that can be achieved when split is allowed in performing time capacitated tasks. When time capacitated tasks are considered, there are, however clear differences in the type of savings. As in VRP setting, the major objective is to decrease the number of vehicles and the length of total distance, in time capacitated resource allocation the major objective can be to minimize the number of resource time needed to complete all tasks. As in transportation business, it may be possible to lease a vehicle for a half a day only, in service setting the regulations or company policy may rule hiring workers for complete working days only. A worker's slack time may depend on whether he is paid also for the time when he is idle or only for the time he is working. Consequently, savings achieved by the time capacitated task splitting can not be determined without considering different aspects of business. By small and simple examples, we can, however, find examples that show high "relevant" savings potential.

4.1. 40% Savings in Capacity Time Shown by an Easy Example

Let's assume three tasks with one hour requirement, zero distances and three workers whose capacities are C_1 =]1, 2[, C_2 =]1, 2[, $C_3 = 2 - \varepsilon$, and $C_1 + C_2 = 3$. In this case, not allowing splits requires each worker to be allocated to one task only. None of the workers has enough capacity to complete two tasks. Because there are only 3 hours of work but $5 - \varepsilon$ hours of capacity, we have $1 - 3/(5 - \varepsilon) \approx 40\%$ slack. If we allow task splitting, we can avoid slack completely because two workers with the total capacity of three hours can complete all three tasks. 40% savings in capacity can be reached. This example is illustrated in Figure 8.

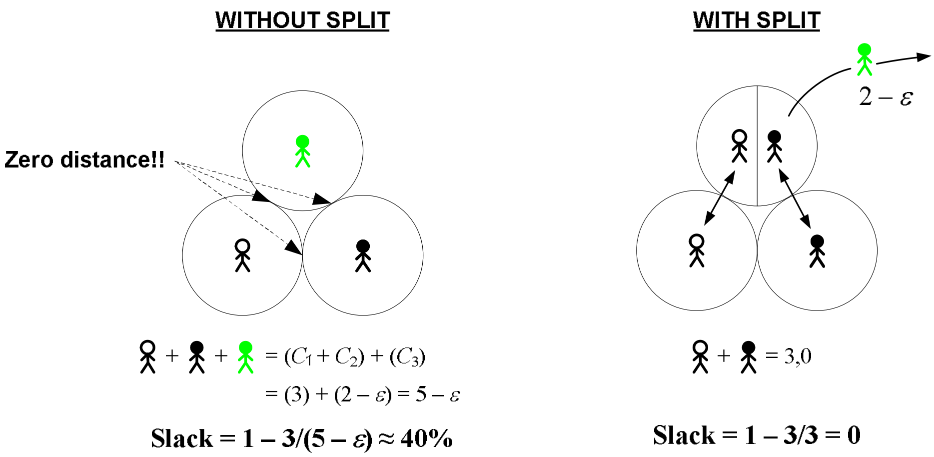

Figure 8. 40% savings by task splitting

4.2. Two Ways of Showing 50% Savings in Capacity Time

The First Way

The possibility to achieve up to 50% savings by task splitting can be easily seen by observing different solution cases for a problem with two tasks and two workers. A worker has slack is its capacity is not completely used and the worker can not be, in practice, removed from the solution.

Let's have

- two tasks $i \in \{1, 2\}$
- zero distance $d_{12} = 0$ between tasks
- one hour of work required by both tasks $R_1 = R_2 = 1$
- two workers whose capacities are $C_1 = C_2 = 2h - \varepsilon$, $C_1 + C_2 = 4 - 2\varepsilon$, and ε is a small number

Let $h_{p,i}$ be the slack of resource $p \in \{1, 2\}$ in task $i \in \{1, 2\}$ and H be the total slack of resources $p \in \{1, 2\}$.

Assume that ε is too big to be rounded to zero by a computer but small enough to have any significance in real life. This example is illustrated in Figure 9.

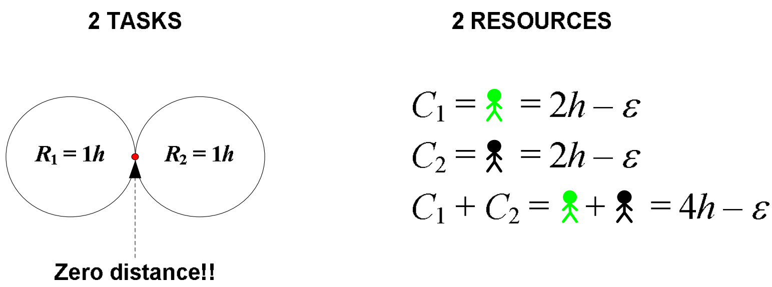

Figure 9. Savings potential example

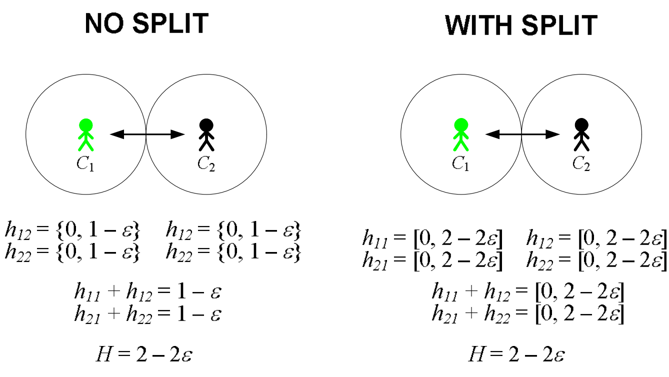

Figure 10. Work allocation when splits are allowed and when they are not

For this example we have two solutions that are illustrated in Figure 10. If we do not accept split, both workers work one hour and they have $1 - \varepsilon$ hours of slack. Altogether there is $1 - 2/(4 - 2\varepsilon) \approx 50\%$ of slack as long as ε can not be rounded to zero. Only, if $\varepsilon = 0$, one worker can be removed from the solution because the other worker can do the both tasks.

As $\lim_{h_{11}+h_{12} \to 2}$ or $\lim_{h_{21}+h_{22} \to 2}$, the work load of one worker increases and the work load of the other worker decreases as in Figure 11. Because there are only two tasks, we can not efficiently use the excess capacity of the two allocated workers. If also $\lim_{\varepsilon \to 0}$ then the final stage, in practice, is that the other worker is no more needed and we get rid of 50% the original capacity.

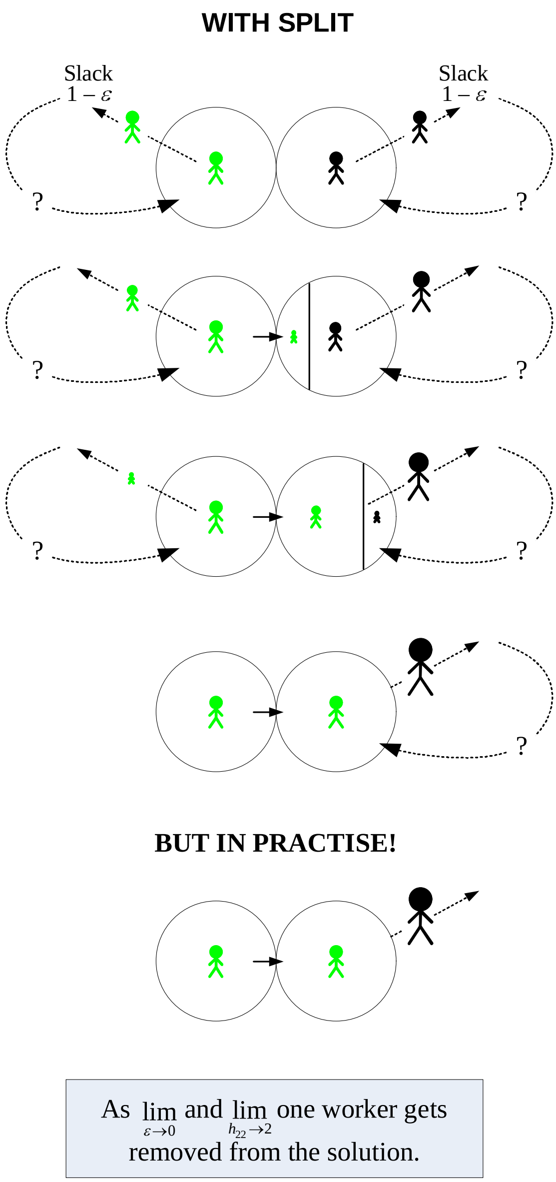

Figure 11. As splits are allowed, the slack per worker vary between 0 and 2 – 2ε.

The Second Way

We can also use the previous example to strictly prove the 50% savings potential of task splitting by adding a third worker whose capacity C3 = ε. If we have $C_1 = C_2 = 2h - \varepsilon$, $C_3 = \varepsilon$, and $C_1 + C_2 + C_3 = 4$, the solution without splits requires $C_1 + C_2 = 4 - 2\varepsilon$ when the solution with splits can be achieved by $C_1 + C_3 = 2$ or $C_2 + C_3 = 2$. With this approach the saving in capacity is, $2 – 2\varepsilon$ which again is practically 50% of the original capacity if ε is small.

5. Time capacitated Resource Allocation Problem with Split Tasks (TCRAPST)

This chapter describes the modeling of Time Capacitated Resource Allocation Problem with Split Tasks (TCRAPST). In the TCRAPST, all tasks are defined by their location and time required to perform the tasks. Also resource capacity is measured as time. Resources can be moved between different tasks and this transportation time consumes resource capacity. Let us call this moving time set-up time. (Unproductive time between tasks actually consists of at least four components: (1) set-down time, (2) time spent on moving, (3) potential slack and (4) the actual set-up time.) As we measure resource capacity as time, we get an answer to the question how that capacity time is allocated between tasks, set-ups and slack.

All resources are assumed to be equally efficient and their capacities predefined. An hour task performed by anyone of the resources will thus consume one hour of resource’s time. Based on these assumptions, the total time spent on performing all tasks is known. The length of a task can exceed the length of a planning period leading to a need for task splitting because the task exceeding the length of a planning period can not be completed by one resource only. However, also tasks shorter than a planning period can be split between resources. The TCRAPST can handle both cases.

The objective of the TCRAPST is to minimize the total set-up time between tasks. At the same time, slack time gets maximized. By maximizing slack, we can learn how much overcapacity we have. As we know our overcapacity, we can make decisions either to use less capacity or to find new tasks where overcapacity can be used.

In the TCRAPST, resources are modeled separately but interconnected by capacity constraints and resource time required by customers. So, if one resource is not enough for completing a task, the TCRAPST automatically allocates more resources for that task.

In the TCRAPST traveling time from task i to task j is assumed to be the same as from task j to i allowing the use of undirected set-up time variables between tasks.

Figure 12 illustrates how a modeler sees a TCRAPST solution as paths. The paths start and end with a virtual depot with zero set-up times for set-ups entering and leaving the depot. User of the solution does not see the virtual depot. Instead, for a user the solution is shown as paths that start from a task, go via tasks and end in tasks. Paths are not directed which means that either end of a path can be a starting point. A path can also be one task long which means that a resource is employed in one task only. If there is a real base in real-world application where resources are located when they are not working, the base (or depot) and routes leaving and entering it can be made visible and given actual transportation times.

In Figure 12, a working shift of four resources is modeled as paths. One task is shared by two resources, so two paths (Path 1 and Path 2) have one common task. One task on Path 3 is big enough to consume the whole capacity of the third resource. Path 4 has three tasks and does not have any shared tasks with other paths.

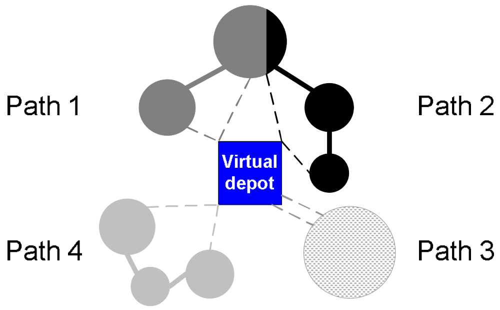

Figure 12. Illustration of paths generated by the TCRAPST

Drawn as figures, the TCRAPST solutions, especially if set-up times are traveling times between differently located tasks, resemble solutions to Vehicle Routing Problems with Split Deliveries (SDVRP). In Vehicle Routing Problems (VRP) vehicle capacity is used to deliver products or people. In the TCRAPST, we deliver service or production capacity measured as time. Personnel rostering and scheduling problems also have much in common with the TCRAPST but they are not considered here.

5.1. MILP formulation of the TCRAPST

Notations

Sets

P	= set of resources	
I	= ordered set of tasks	

Parameters

$D_{i,j} = D_{j,i}$	= set-up time between tasks i and j,	$i \in I, j \in I$
R_i	= capacity required by task i,	$i \in I$
C_p	= capacity of resource p,	$p \in P$
K_p	= cost of using resource p,	$p \in P$
M	= big number	

Binary variables

$s_{p,i}$	= 1, if resource p visits task i; 0 otherwise,	$p \in P, i \in I$
$d_{p,i,j} = d_{p,j,i}$	= 1, if resource p moves between tasks i and j; 0 otherwise,	$p \in P, i \in I, j \in I$
$b_{p,i}$	= 1, if resource p starts the period in task i; 0 otherwise,	$p \in P, i \in I$
$e_{p,i}$	= 1, if resource p ends the period in task i; 0 otherwise,	$p \in P, i \in I$

Continuous variables

$w_{p,i}$	= capacity time (or work) of resource p used in task i,	$p \in P, i \in I$
h_p	= slack time of resource p,	$p \in P$.

Objective function

$$\min \quad \sum_{p \in P} \sum_{i \in I} \sum_{\substack{j \in I \\ i<j}} d_{p,i,j} D_{i,j} K_p + \sum_{p \in P} \sum_{i \in I} w_{p,i} K_p \qquad (1)$$

The objective function attempts to minimize the number of active resources and the total set-up time simultaneously. The most important objective is to minimize the number of active resources. An active resource here refers to a resource that is used for work.

The TCRAPST allocates all resources to at least one task. However, an allocation to just one task does not require either traveling or work. So, a resource allocated to just one task can either be an active or a non-active resource.

When all resources are allocated, but all allocated resources do not necessarily either work or travel, the cost of using resources for work ($w_{p,i}K_p$) or traveling with different K_p for each resource minimizes the number of active resources needed to perform tasks. We can, for example, use costs $K_{p+1} = 10K_p$ which uses cheapest resources first and forces more expensive resources on paths with no traveling or allocated work. Simultaneously, we maximize the slack of the most expensive resource that is actually used.

The constraints of this model can be grouped into capacity constraints and set-up constraints. Capacity constraints regulate the distribution of resource capacity time between work, set-ups and slack. Set-up constraints define the possible routes that can exist for each resource between tasks.

Capacity constraints

$$\sum_{p \in P} w_{p,i} = R_i \qquad , \forall\, i. \qquad (2)$$

$$\sum_{i \in I} \left(w_{p,i} + h_p \right) + \sum_{i \in I} \sum_{\substack{j \in I \\ i<j}} d_{p,i,j} D_{i,j} = C_p \qquad , \forall\, p. \qquad (3)$$

$$w_{p,i} \leq s_{p,i} M \qquad , \forall\, p \text{ and } i. \qquad (4)$$

Constraint (2) requires that all work in tasks gets done. According to constraint (3) time spent on work, set-ups, and slack has to equal the capacity of a resource. Because a resource has to be physically available to do its task, in constraint (4) $s_{p,i}$ is 1 for each resource p working in a task i during a period and 0 otherwise. This formulation allows a resource to be routed via one task only and without any working on its path. Resources without any allocated work can be considered as inactivated, and thus unneeded, resources.

Set-up constraints

$$\sum_{i \in I} b_{p,i} = 1 \qquad , \forall\, p. \qquad (5)$$

$$\sum_{i \in I} e_{p,i} = 1 \qquad , \forall\, p. \qquad (6)$$

$$s_{p,i} - \frac{\left(b_{p,i} + e_{p,i} \right)}{2} - \frac{\left(\sum_{\substack{j \in I \\ i<j}} d_{p,i,j} + \sum_{\substack{j \in I \\ j<i}} d_{p,j,i} \right)}{2} = 0 \,, \forall\, p \text{ and } i. \qquad (7)$$

$w_{p,i}, h_p \geq 0$, $\forall$ p, and i,

$b_{p,i}, d_{p,i,j}, e_{p,i}, s_{p,i} \in \{0,1\}$, $\forall$ p, i, and j.

Constraints (5) and (6) forces there to be one first and one last task for each resource p. Constraint (7) lets set-ups or movements only take place between tasks where the decision of working has been made.

Constraint (7) also requires for each resource at least one allocation in any task even when we do not actually need that resource.

If we want, we can add a clarifying, but redundant, constraint (8).

$$0 \leq \sum_{i \in I} s_{p,i} - \sum_{i \in I} \sum_{\substack{j \in I \\ i<j}} d_{p,i,j} \leq 1 \qquad , \forall\ p. \qquad (8)$$

which forces that a resource p can arrive only once and depart only once from a task i. If resource p works in more than one task during a planning period, there has to be a set-up between the tasks. Constraint (8) also forces these set-ups: A worker working in two tasks has to travel once; a worker with three tasks travels twice etc.

As the model was tested with generated data the solution did not always behave as we would expect of the objective function (1). Strange solutions are probably caused by scaling and rounding that CPLEX 9.0 does during the solution process because the difference in objective function value between strange solution and intuitively optimal solution is typically very small.

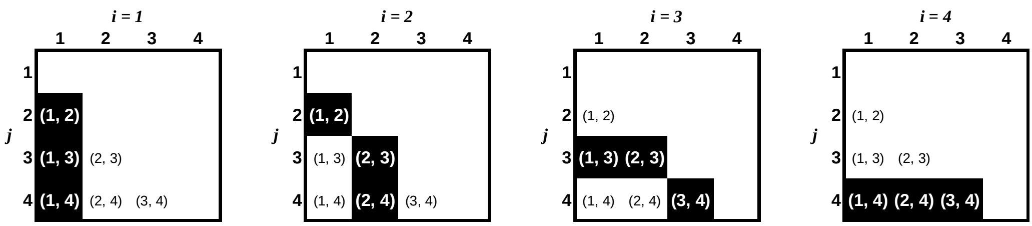

Figure 13. Possible combinations for each $i \in (1, 2, 3, 4)$

To illustrate the logic of constraint (7), let's assume 4 tasks in a system and tasks presented as an ordered set. Numbers in both axes are tasks. Combinations $i = j$ are not allowed. The upper right hand side of the matrix is not needed because the set is ordered.

Let the darkened cells in Figure 13 show the possible combinations for each $i \in (1, 2, 3, 4)$. If we further assume that the decision of working in task 3 has been made for resource p, constraint (7) allows $d_{p,1,3} + d_{p,2,3} + d_{p,3,4} \leq 2$. The (i, j)-combinations allowed for variables $d_{p,i,j} = d_{p,j,i}$ $(i \neq j)$ and $s_{p,3} = 1$ are listed in Figure 14. The combinations in the first row of 3 are possible when the task $i = 3$ is not the first or the last task on resource's path. The combinations in the second row of 3 are possible when the task $i = 3$ is either the first or the last task allocated for a resource on its path.

Constraint (7) allows the maximum of one arrival to and one departure from a task i for an allocated resource p, but no arrivals or departures when $s_{p,i} = 0$. For $I = 4$ and $i = 3$ or $j = 3$ Table 4 shows the existing possible transfer combinations.

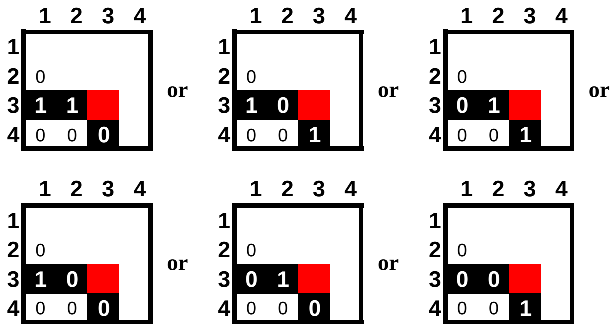

Figure 14. Allowed binary combinations for $I = 4$, $i < j$ and $i = 3$

Table 4. Possible binary combinations for $I = 4$, $i < j$ and $i = 3$

$s_{p,3}$	$d_{p,1,3}$	$d_{p,2,3}$	$d_{p,3,4}$
1	1	1	0
1	1	0	1
1	1	1	0
1	0	1	1
1	1	0	1
1	0	1	1
0	0	0	0

5.2. Subtour Considerations

Subtour is a closed route that does not have a beginning or end. In the MILP formulation of The TCRAPST unwanted subtours can exist if resources are allowed to work in more than three tasks within a planning period. This feature seriously limits the applicability of the formulation because the elimination of subtours by additional constraints makes the model even more difficult to solve and construct.

Lines in Figure 15 illustrate the paths of a potential solution for a system with two resources and six tasks ($P = 2$ and $I = 6$). Resource 2 has a desired path that starts from outside the system and exists to outside of the system. However, resource 1 has two paths of which one is an undesired subtour.

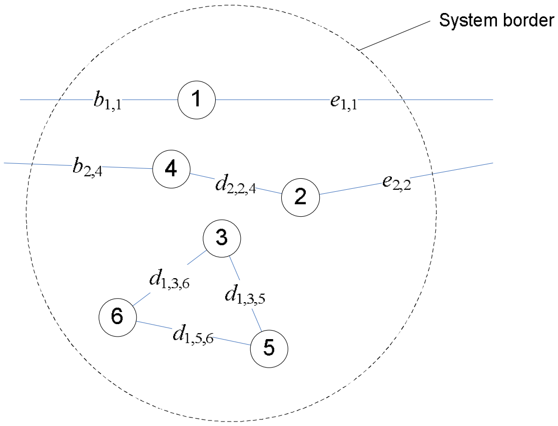

Figure 15. Subtours can exist in MILP formulation of the TCRAPST if a resource is allowed to work in more than 3 tasks within a planning period

We can completely avoid subtours by limiting the number of allowed tasks for a resource within a planning period to the maximum of three. If we add constraints (9) and (10), a resource can work in up to four tasks within a planning period without subtours. Because constraint (9) determines for a resource that it can not start and end its path in the same task, we automatically select out one-task-paths. With the maximum of 4 tasks per path this also forces out three-task-subtours because at least two tasks out of four have to have a movement in or out of the system.

$$b_{p,i} + e_{p,i} \leq 1,\ \forall\ p \text{ and } i \tag{9}$$

$$\sum_{i \in I} s_{p,i} \leq 4 \qquad ,\ \forall\ p \tag{10}$$

Constraints (5), (9) and (10), together require for each resource p that $\sum\sum d_{p,i,j} \geq 1$ which means that each resource has be allocated to at least two tasks even when one task would be big enough to consume all capacity of a resource. Constraints (5) and (6) also require that all available resources have to be allocated to paths even when some resources have zero capacity. Together with constraint (9) they actually require each resource to go through two tasks.

If constraints (9) and (10) are not used, another way to avoid subtours is to add constraints to specifically exclude each possible subtour. For example, a subtour consisting of three tasks *a*, *b*, and *c* can be excluded by constraint (11). Similarly, a 4-task subtour consisting of tasks *a*, *b*, *c*, *d* can be avoided by constraint (12). The number of subtours constraints increases very fast as bigger and bigger subtours are to be avoided making the enumeration of all possible subtour constraints an impractical approach.

$$d_{p,a,b} + d_{p,a,c} + d_{p,b,c} \le 2 \qquad , \forall\ a, b, c \in I \text{ and } a<b<c. \qquad (11)$$

$$d_{p,a,b} + d_{p,a,c} + d_{p,a,d} + d_{p,b,c} + d_{p,b,d} + d_{p,c,d} \le 3,$$
$$\forall\ a, b, c, d \in I \text{ and } a<b<c<d. \qquad (12)$$

Limiting a problem to include the maximum of four tasks on a path can be a good strategy for example in allocating cleaning personnel. If it shows up that tasks are shorter than 1/4 or 1/3 of a planning period, we can try to preprocess and simplify a problem by combining short tasks with nearby tasks before we send the problem to a solver.

Probably the easiest way to avoid subtours is to combine tasks in a preprocessing stage and present them to the model as one bigger task. By this way the number of tasks presented for the optimization model can be decreased and their length increased up to the point where the number of tasks a resource can visit within a planning period becomes automatically limited.

We can also use an approach where we first start solving a problem without subtour constraints and then add those constraints one by one to avoid only those subtours that emerge. Because after adding new constraints the problem has to be solved again, this approach is also inefficient. However, in that way we can generate optimal reference solutions that can be compared with solutions generated by other methods.

Salkin (1989, 17) describes a subtour prevention constraint $a_i - a_j + (n+1)d_{p,i,j} \le n$ used in the Traveling Salesman Problem (TSP). In that TSP-constraint a_i is a real number associated with task i and n is the number of tasks allowed for a round. Unfortunately, this subtour prevention constraint can not be used here because we do not know it in advance, how many tasks a resource is allocated to within a planning period.

Subtours in the TCRAPST could possibly also be avoided by using directed arcs and suitable constraints.

5.3. Fixed Resource Activation Cost

Objective function (1) does not charge any fixed cost on the allocation of a resource to just one task without any allocated working time on that resource. All resources are routed through at least one task. Inactivated resources just go through one task but they do not work. A path of an inactivated resource includes one task but no work because entering a path ($b_{p,i}$) and leaving a path ($e_{p,i}$) are not charged with any cost.

We can introduce a fixed cost of activating a resource by including two sets of tasks, active tasks and passive tasks. Active tasks include all real tasks where work hast to be done. The set of passive tasks includes only one task δ ($\delta \in I$ and $R_\delta = 0$) without a working requirement.

Now we can allocate a fixed cost on starting a path in an active task ($b_{p,i}$, $i \neq \delta$) and leave a path starting in a passive task ($b_{p,\delta}$) uncharged. If a resource is not activated, it will now be routed through the task δ only because all other paths incur costs.

Unfortunately, experiments with the fixed cost formulation were not promising. The TCRAPST formulation allows subtours, and subtour prevention constraints are only added after analyzing each solution. The TCRAPST also allows solutions where a resource works only a very short time in just one task without moving anywhere. The reason for using strongly different capacity usage time costs between resources is that they help to minimize the number of resources working just a very small fraction of their time and being idle the rest of the planning period. As capacity usage time costs are very different between resources, the cheap resources tend to work with their full capacity and the very expensive resources do not work at all. There are not many active resources having substantial slack.

The introduction of a high fixed cost of activating a resource leads to a situation where the TCRAPST first tries a large number subtours before activating an additional resource. As the number of subtour constraints increases, the problem often finally becomes so constrained that CPLEX fails to find a new feasible initial solution. Even when a feasible and reasonably good solution could finally be found, the creation of numerous subtour constraints after each resolving makes the solution process to last impractically long.

5.4. Lower Bound

The TCRAPST solves to optimality only with very small problems having about 10 tasks and 10 resources. In a case of a bigger problem, one would be interested in knowing how far away from the optimal solution the found non-optimal solution is. The optimal solution is somewhere between the existing feasible non-optimal solution and a lower bound which is a solution to a relaxed problem. Loosening or removing a constraint and dropping a binary requirement are typical relaxations. A good lower bound is computed fast and it does not deviate much from the optimal solution to the original problem.

When solving the TCRAPST, we want to find information on the amount of time each resource spends on working and traveling. The binary logic of the TCRAPS determines traveling times based on working time allocations. Changing anyone of the binary variables into a continuous variable destroys the logic resulting in lower bound solutions without allocated traveling time.

The tightness of this lower bound depends on the relative distances of tasks compared with task lengths. If task lengths are long compared with distances between tasks, the lower bound solution is tighter than when the relative distances between tasks are longer.

When relaxing the TCRAPST, we may want to preserve as much of the problem structure as possible. We could, for example, remove subtour constraints and get lower bounds quite close to the optimal solutions of the original problems. However, removing subtour constraints does not make the problem much easier and faster to solve because the number of binary combinations to consider still stays high.

Another very fast solving lower bound retaining the binary logic can be found by changing constraints (5) and (6) to $\sum b_{p,i} \leq M$ and $\sum e_{p,i} \leq M$ with $M \geq I$. This relaxation allows every resource to take as many as I paths. By this relaxation, we get a lower bound consisting of just working time. There is no traveling because all paths consist of one task only. We can tighten this lower bound, with the penalty of longer solution time, by decreasing the M, which will incrementally introduce more and more traveling between tasks.

6. TCRAPST Solution Example

To illustrate a TCRAPST solution we could provide a solution to the question below:

> *How to allocate 25 cleaning people within an 8 hour work shift to 25 tasks with different durations so that the number of people actually needed is minimized?*

Figure 16 illustrates the problem with the centers of circles corresponding to task locations, the diameters of circles to task lengths, and euclidean distances between the centers of circles to traveling times between tasks. Distances between the centers of circles correspond to the distance hour scales of the axes. The diameters of circles do not directly relate to axes. A big circle is a long task and a small circle is a short task. For this example, the lengths of tasks are randomly generated from the normal distribution N[1,8]. Horizontal and vertical coordinates are randomly generated from the normal distribution N[0,5]. The resulting task lengths and coordinates are listed in Table 5.

Table 5. Task lengths and coordinates for the example problem

Task number	1	2	3	4	5	6	7	8	9	10	11	12	13
Horizontal	0.01	4.04	1.75	3.73	3.55	0.07	0.74	2.23	0.04	2.86	0.83	1.76	3.92
Vertical	2.82	2.93	4.48	0.87	2.57	0.46	0.83	0.60	1.89	3.01	3.32	0.29	4.01
Task length	2.35	4.36	6.76	7.01	3.13	3.55	7.92	1.03	4.72	5.25	4.16	5.25	4.64

Task number	14	15	16	17	18	19	20	21	22	23	24	25
Horizontal	1.51	4.78	0.71	4.31	4.22	3.06	1.49	1.88	0.28	1.38	3.46	2.42
Vertical	4.38	4.63	2.31	1.05	4.98	1.96	4.20	0.46	0.04	1.36	4.19	1.03
Task length	6.09	4.78	2.65	6.46	8.00	2.86	1.17	5.74	7.43	5.12	6.09	6.21

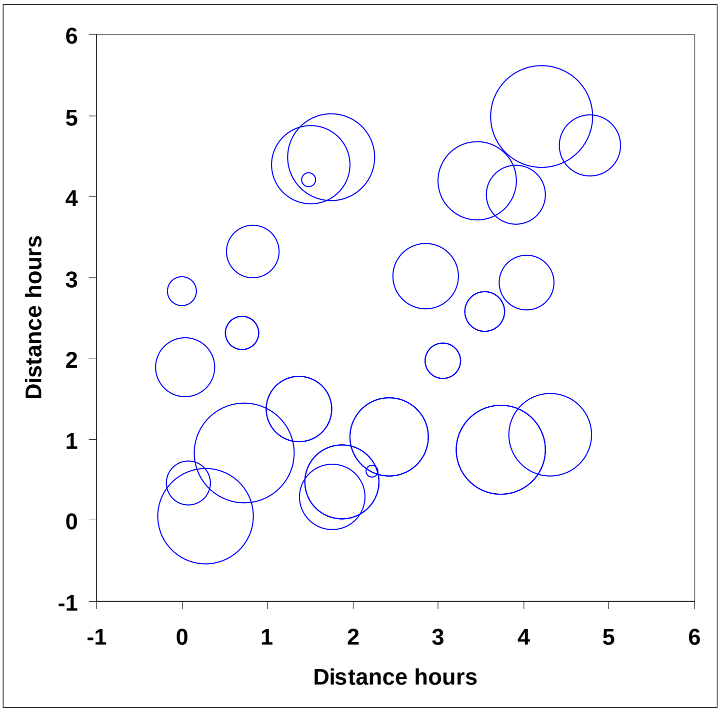

Figure 16. Tasks scattered on a "map"

The cost of time spent on traveling was the same cost for each worker, but, for each worker, the working time in tasks was given a different cost so that the working time of a worker, or the time spend in tasks, increases as workers' ordinal number increases, i.e. worker number one is the cheapest to use and worker number 25 is the most expensive to use. As the capacity cost of different workers varies, the solver tries to find solutions where least expensive workers are used and the use of most expensive workers is avoided.

Figure 17 and Figure 18 illustrate solutions generated by the TCRAPST. In Figure 17, the diameters of circles illustrate the length of tasks. In Figure 18, all tasks are illustrated with same-sized small circles to highlight the paths allocated to workers. Circles filled with black point out the tasks that are split between different workers.

Due to problem complexity, these solutions are only indicative, not optimal. This can also be seen in Table 5 where working time of workers does not always decrease as worker's ordinal number increases. In an optimal solution, the utilization of cheaper workers in tasks should be higher than the utilization of more expensive workers. For example, by exchanging the work loads of the worker 13 and the worker 14 we could manually improve the objective function value. Still, even this non-optimal solution is useful because it illustrates typical solution features.

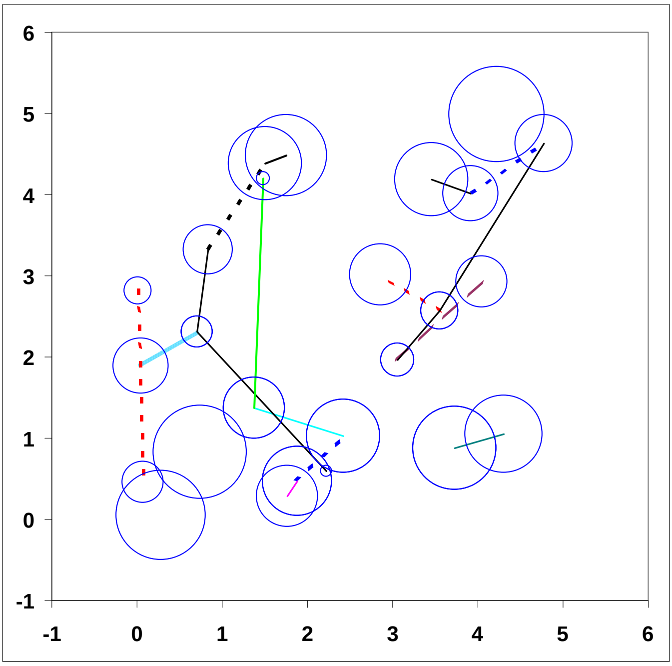

Figure 17. Solution generated by the TCRAPST

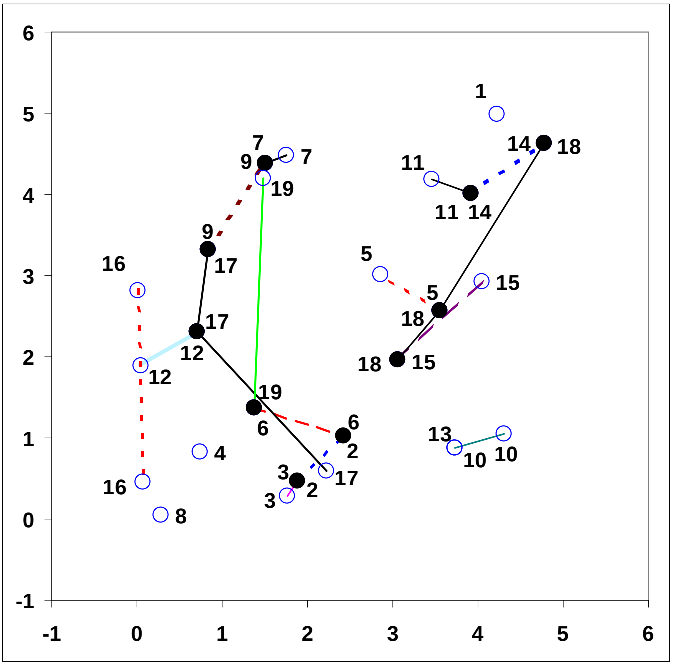

Figure 18. Workers allocated to tasks in solutions generated by the TCRAPST

In Figure 18, only the locations of tasks are shown but the length of tasks is ignored. In the TCRAPST solution shared tasks are colored with black. The numbers indicate workers working in each task. Table 5 shows the capacity time usage of each worker. Altogether 19 workers are needed to complete 25 tasks. Workers 1 – 15 all spend at least 79% of their time in actual work. Then proportional time of workers in tasks starts decreasing rapidly. The last activated worker has the largest slack of all activated workers. Table 6 and Table 7 indicate workers allocated for each task. In the TCRAPST solution, 19 workers are needed as 10 tasks are shared by more than one worker.

Table 5. Working times, traveling times, utilization for tasks and slack

Worker	Working time	Traveling time	Working + Traveling	Utilization for tasks	Slack
1	8.00		8.00	100 %	
2	6.94	1.06	8.00	87 %	
3	7.45	0.55	8.00	93 %	
4	7.92		7.92	99 %	1 %
5	6.93	1.07	8.00	87 %	
6	6.82	1.18	8.00	85 %	
7	7.41	0.59	8.00	93 %	
8	7.43		7.43	93 %	7 %
9	6.67	1.33	8.00	83 %	
10	7.13	0.87	8.00	89 %	
11	7.20	0.80	8.00	90 %	
12	6.87	1.05	7.92	86 %	1 %
13	6.34		6.34	79 %	21 %
14	6.78	1.22	8.00	85 %	
15	6.60	1.40	8.00	83 %	
16	5.90	1.56	7.46	74 %	7 %
17	4.46	2.87	7.33	56 %	8 %
18	3.60	2.87	6.47	45 %	19 %
19	2.28	1.72	4.00	28 %	50 %

Average capacity used in tasks	81 %
Average capacity spent on traveling	13 %
Average slack per worker	6 %

Table 6. Tasks allocated per worker

Task number	1	2	3	4		5		6	7	8
Worker	16	15	7	10	13	5	18	16	4	17
Working time per task	2.35	4.36	6.76	0.67	6.34	1.68	1.45	3.55	7.92	1.03
Total work in task	2.35	4.36	6.76	7.01		3.13		3.55	7.92	1.03

Task number	9	10	11		12	13		14	
Worker	12	5	9	17	3	11	14	7	9
Working time per task	4.72	5.25	1.23	2.93	5.25	1.11	3.53	0.65	5.44
Total work in task	4.72	5.25	4.16		5.25	4.64		6.09	

Task number	15		16		17	18	19		20
Worker	14	18	12	17	10	1	15	18	19
Working time per task	3.25	1.53	2.15	0.5	6.46	8.00	2.24	0.62	1.17
Total work in task	4.78		2.65		6.46	8.00	2.86		1.17

Task number	21		22	23		24	25	
Worker	2	3	8	6	19	11	2	6
Working time per task	3.54	2.20	7.43	4.01	1.11	6.09	3.40	2.81
Total work in task	5.74		7.43	5.12		6.09	6.21	

Table 7. Workers allocated per task

Worker	1	2		3		4	5		6	
Task number	18	21	25	12	21	7	5	10	23	25
Working time per task	8.00	3.54	3.40	5.25	2.20	7.92	1.68	5.25	4.01	2.81
Total working time	8.00	6.94		7.45		7.92	6.93		6.82	

Worker	7		8	9		10		11	
Task number	3	14	22	11	14	4	17	13	24
Working time per task	6.76	0.65	7.43	1.23	5.44	0.67	6.46	1.11	6.09
Total working time	7.41		7.43	6.67		7.13		7.20	

Worker	12		13	14		15		16	
Task number	9	16	4	13	15	2	19	1	6
Working time per task	4.72	2.15	6.34	3.53	3.25	4.36	2.24	2.35	3.55
Total working time	6.87		6.34	6.78		6.60		5.90	

Worker	17			18			19	
Task number	8	11	16	5	15	19	20	23
Working time per task	1.03	2.93	0.5	1.45	1.53	0.62	1.17	1.11
Total working time	4.46			3.6			2.28	

7. Extensions of the TCRAPST and Discussion on Business Applications

7.1. Extensions of the TCRAPST

Many important aspects were ignored in the previous example to keep it simple. Can we really assume that all workers, for example, are similar and equally efficient? Will there be congestion in shared tasks that decreases the efficiency of workers? Can savings from the decreased number of workers be lost by the increased costs of traveling? Is it economical to allocate workers for very short periods in tasks? For example, in Table 6 we see that four workers [7, 10, 17, and 18] were allocated to work less than one hour in a one of their tasks. In real life that would probably not make sense.

Luckily, the TCRAPST assumptions can easily be relaxed. Excessive traveling and very short working times in tasks can be discouraged by additional constraints. Different efficiencies and different travelling speeds of workers can be handled by parameters. Traveling times can also be expressed as set-up times not directly dependent on actual distances. In that way, the model can be applied to a production environment where workers capable of working in several tasks do not actually move long distances but there is still a considerable delay between ending one task and starting another task. Different skills can be taken into account by limiting workers' allocation to tasks. We can also replace the undirected transfer network with a directed one to include time windows and interconnected time periods.

In the TCRAPST, a virtual depot was used, but we can also use a real depot or a base where people start their work shift and where they come back after a work shift. More than one base can be included, as in Figure 19, and the model can be used to allocate workers to tasks and customers to bases.

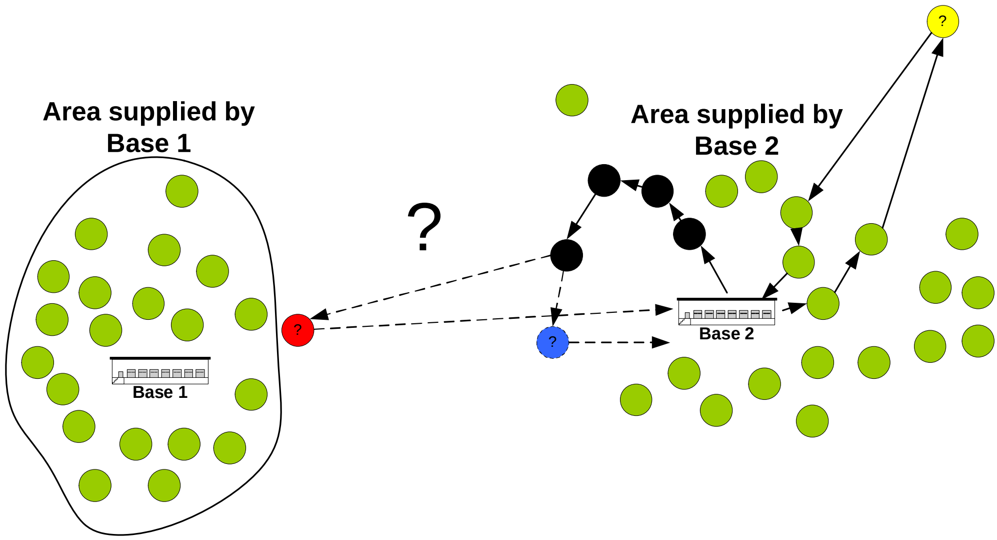

Figure 19. Example of a problem with two bases

More detailed modeling is not the biggest problem with the TCRAPST. The real problem is the computational complexity of the problem. Therefore, more efficient solution methods are needed to put the TCRAPST in practice.

7.2. Business Application

Time capacitated resource allocation can be applied in services where tasks are similar and resources, such as cleaning personnel, have similar and standardized skills and efficiencies. Standardization of tasks and skills makes it possible to make reliable forecasts about the capacity consumption of tasks as well as efficiencies of resources. If tasks, skills and efficiencies can not be forecast accurately enough, time capacitated resource allocation should not be used.

One application of the TCRAPST is to use it as a decision support tool when analyzing company's customer base. Figure 20 illustrates a cluster, a suburb for example, with four people serving 13 customers. The current optimal paths of each service person are marked with colors. Based on our current capacity usage and costs, we consider a potential new customer A.

We can use the TCRAPST together with cost accounting to assist us in answering different questions. If there is only additional customer A to be considered, we could ask: Should customer A be included in our customer base? Do we have enough capacity to serve customer A? Which one of our service people should serve customer A? How much, and on what cost, should we increase our capacity that it is profitable for us to serve customer A? Instead of hiring a new worker, could we handle the additional capacity need by applying overtime? Should our competitor serve customer A instead of us? What price should customer A be served for? Can and should we use price differentiation? Does a plan suggested by the TCRAPST make common sense?

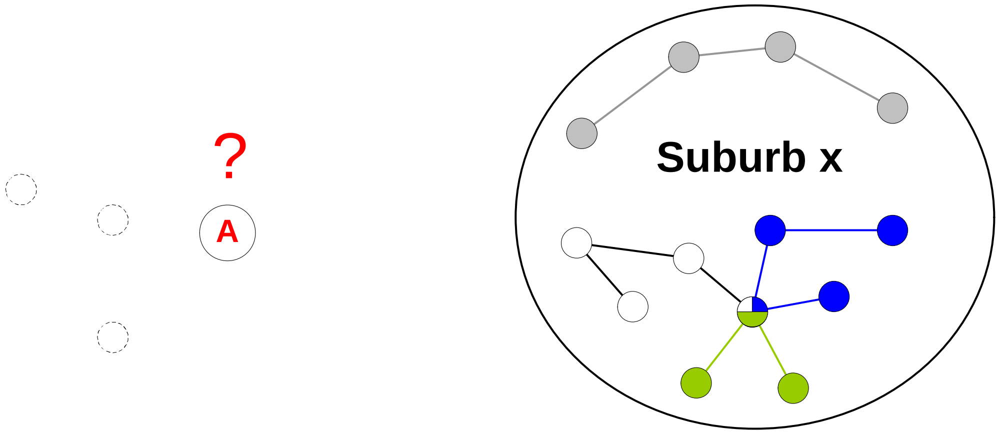

Figure 20. Example with 13 existing customers, service paths and potential new customers

If there are other potential customers nearby customer A, we can ask additional questions. Should we now be interested in customer A? Should we increase our capacity to serve customer A and customers nearby? What reallocations of our workers would that require? Should we make a special offer to get customer A and its nearby customers into our customer base? How many new customers do we want?

Rostering and scheduling of staff is a typical problem and it has been studied extensively in nurse scheduling, air craft scheduling etc. Typically, the problem is to solve a matching problem where one person is matched with each task. The splitting of tasks between more than one person is seldom considered and having that split as a decision variable hardly never.

However, there are tasks where considering the splitting of tasks time between different resources as a decision variable can be beneficial. A typical example is the staff allocation of a cleaning company having a set of buildings to be cleaned. I such a case, the cleaning of a building, the cleaning of different floors, and time needed to move between different tasks can be considered as tasks for which duration can be forecast. Based on the time forecast, the tasks can be allocated for cleaning staff. Some people may work the whole day with the same task. Other people may have to move between several tasks that may be completed by one person or be split to be completed by more than one person. The objective for a planner in that kind of a case is to allocate and schedule personnel in such a way that traveling and other kind of time not adding value is minimized.

Other examples where splitting can be beneficial are the allocation of expensive leased equipment, staff planning on the factory floor with flexible and multi-skilled work force as well as in many services such as home health care, facility maintenance, and private security services. Despite modeling and optimizing the simultaneous movements of several people taking different time consuming activities (working, set-up, set-down, transfer, and idle time) and flexible job splitting into account offers potential for substantial savings, the subject has not received much attention in literature. To my knowledge, the only relevant research closely related can be found in vehicle routing literature.

As time is used as constraint, the ability to measure and forecast durations is a key. Forecasting is easier as tasks are standardized and repetitive. If the duration of tasks and traveling can not be measured, time capacitated optimization becomes meaningless.

In practice, however, experienced people and machines typically have quite stable production rates that can be measured. As people get more experienced, they typically become faster. Experience increases as similar tasks are repeated again and again. Learning can continue in different locations if tasks are similar. So, if we allocate an experienced person to a typical task, we can quite accurately estimate how long it takes to complete the task. The same applies for many other resources.

8. Conclusions

Time capacitated resource allocation can be applied in services where tasks are similar and resources, such as cleaning personnel, have similar and standardized skills and efficiencies. Standardization of tasks and skills makes it possible to make reliable forecasts about the capacity consumption of tasks as well as efficiencies of resources. If tasks, skills and efficiencies can not be forecast accurately enough, time capacitated resource allocation should not be used.

This essay described a resource allocation model that can be used when both capacity and requirements are expressed as time. That MILP model helps to allocate resources to perform tasks so that the number of resources gets minimized. In minimization of the number of resources needed, the routing of resources is a key because both working in tasks and moving between tasks also consume capacity.

In addition to routing, task splitting was taken into account. As resources and tasks are very similar, it is often practical to let more than one resource to work in a task, especially, if a task takes a long time complete. The MILP model presented in this essay also does that splitting of tasks between resources when simultaneously minimizing the number of resources needed in the whole system.

The focus on this essay was on savings potential of task splitting in time capacitated modeling. Firstly, examples on that savings potential was given by comparing solutions that do not allow task splitting to solutions that include task splitting. Secondly, I was proven that task splitting can, in a theoretical case, bring up to 50% savings in comparison to a solution that does not allow task splitting. Thirdly, a Mixed Integer Linear Programming (MILP) problem model on Time Capacitated Resource Allocation Problem with Split Tasks (TCRAPST) was presented. Finally, the extensions and potential applications of the TCRAPST were discussed.

This essay concentrated on describing and proving a model that can potentially generate more efficient operational plans than existing models. The focus was on savings effect and, therefore, many important practical aspects, such as time windows or minimum working time constraints, were ignored. If split tasks are to be applied in practice, the model has to be extended.

Another subject for further research would be to test the savings effect with data by comparing solutions generated by the TCRAPST with solutions generated by a similar model that does not allow task splitting. Data could also be used to test whether reduced the TCRAPST solutions really are optimal. Does the TCRAPST always allocate complete resource work shifts to tasks when task length exceeds the capacity of resource's work shift?

As a MILP formulation, the TCRAPST could only solve very small problems. If the idea of task splitting in time capacitated problems is to be put in practice, more efficient solution methods have to be developed. When developing those methods, the TCRAPST solutions, or solutions to its extensions, can be used as reference solutions to compare the quality of solutions generated by other techniques than branch-and-bound based optimization. The TCRAPST model can serve as a starting point in analyzing the implications of task splitting, set-ups, reallocations and set-downs in different industries.

Second routing application area where resource capacity can be measured in time is human resource or robot allocation and routing for different tasks.

Then a short survey is made on transportable resource allocation, routing and scheduling in cases where requirements can be measured as capacity time. Based on these surveys it becomes quite obvious that a research gap exists when it comes to modeling the flexible splitting of resource time between tasks.

List of References

Archetti, C., Mansini, R. & Speranza, M.G. (2005) "Complexity and Reducibility of the Skip Delivery Problem", *Transportation Science,* Vol. 39, No. 2, pp. 182-187.

Archetti, C., Mansini, R. & Speranza, M.G. (2001) "The Vehicle Routing Problem With Capacity 2 And 3, General Distances And Multiple Customer Visits", Paris, September 26–29, ORP, pp. 1-8, France.

Archetti, C., Savelsbergh, M.W.P. & Speranza, M.G. (2008) "To split or not to split: That is the question", *Transportation Research Part E: Logistics and Transportation Review,* Vol. 44, No. 1, pp. 114-123.

Archetti, C., Savelsbergh, M.W.P. & Speranza, M.G. (2006) "Worst-Case Analysis for Split Delivery Vehicle Routing Problems", *Transportation Science,* Vol. 40, No. 2, pp. 226-234.

Archetti, C. & Speranza, M.G. (2008) "The Split Delivery Vehicle Routing Problem: A Survey", In: Golden, B., Raghavan, S. & Wasil, E.(eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges,* Operations Research/Computer Science Interfaces Series, Vol. 43, pp. 103-122, Springer, US.

Archetti, C. & Speranza, M.G. (2006) "An Overview on the Split Delivery Vehicle Routing Problem", *Operations Research Proceedings, Department of Quantitative Methods, University of Brescia, Italy,* .

Archetti, C., Speranza, M.G. & Hertz, A. (2006) "A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem", *Transportation Science,* Vol. 40, No. 1, pp. 64-73.

Archetti, C., Speranza, M.G. & Savelsbergh, M.W.P. (2008) "An Optimization-Based Heuristic for the Split Delivery Vehicle Routing Problem", *Transportation Science,* Vol. 42, No. 1, pp. 22-31.

Belenguer, J.M., Martinez, M.C. & Mota, E. (2000) "A Lower Bound for the Split Delivery Vehicle Routing Problem", *Operations Research,* Vol. 48, No. 5, pp. 801-810.

Bompadre, A., Dror, M. & Orlin, J.B. (2006) "Improved bounds for vehicle routing solutions", *Discrete Optimization,* Vol. 3, No. 4, pp. 299-316.

Brenninger-Göthe, M. (1989) *Two Vehicle Routing Problems — Mathematical Programming Approaches,* 1-182p. , Doctoral Dissertation, Linköping Studies in Science and Technology, Department of Mathematics, Linköping University, Sweden.

Campos, V., Corberán, Á. & Mota, E. (2007) *A Scatter Search Algorithm for the Split Delivery Vehicle Routing Problem,* Technical Report, Dep. Estadistica i Investigació Operativa, Universitat de València, Spain.

Chen, S., Golden, B. & Wasil, E. (2007) "The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results", *Networks: an International Journal,* Vol. 49, No. 4, pp. 318-329.

Dror, M., Laporte, G. & Trudeau, P. (1994) "Vehicle routing with split deliveries", *Discrete Applied Mathematics,* Vol. 50, No. 3, pp. 239-254.

Dror, M. & Trudeau, P. (1990) "Split delivery routing", *Naval Research Logistics,* Vol. 37, pp. 383-402.

Dror, M. & Trudeau, P. (1989) "Savings by Split Delivery Routing", *Transportation Science,* Vol. 23, No. 2, pp. 141-145.

Dror, M. & Trudeau, P. (1987) *Split Delivery Routing,* Publication No. 507 (January), Centre de recherche sur les transports, Université de Montréal, Canada.

Eveborn, P., Flisberg, P. & Rönnqvist, M. (2006) "Laps Care—an operational system for staff planning of home care", *European Journal of Operational Research,* Vol. 171, No. 3, pp. 962-976.

Federgruen, A. & Simchi-Levi, D. (1995) "Chapter 4 Analysis of vehicle routing and inventory-routing problems", In: Ball, M.O., Magnanti, T.L., Monma, C.L. & Nemhauser, G.L.(eds.), *Handbooks in Operations Research and Management Science,* Network Routing, Vol. 8, pp. 297-373, Elsevier, North-Holland.

Feillet, D., Dejax, P., Gendreau, M. & Gueguen, C. (2003) *Vehicle Routing with Time Windows and Split Deliveries,* Technical Report, Unpublished.

Frizzell, P.W. & Giffin, J.W. (1995) "The split delivery vehicle scheduling problem with time windows and grid network distances", *Computers & Operations Research,* Vol. 22, No. 6, pp. 655-667.

Frizzell, P.W. & Giffin, J.W. (1992) "The Bounded Split Delivery Vehicle Routing Problem with Grid Network Distances", *Asia-Pacific Journal of Operational Research,* Vol. 9, No. 1, pp. 101-116.

Gendreau, M., Dejax, P., Feillet, D. & Gueguen, C. (2008) *Vehicle Routing with Time Windows and Split Deliveries,* Working Paper, Unpublished.

Ho, S.C. & Haugland, D. (2004) "A tabu search heuristic for the vehicle routing problem with time windows and split deliveries", *Computers & Operations Research,* Vol. 31, No. 12, pp. 1947-1964.

Jin, M., Liu, K. & Bowden, R.O. (2007) "A two-stage algorithm with valid inequalities for the split delivery vehicle routing problem", *International Journal of Production Economics,* Vol. 105, No. 1, pp. 228-242.

Jin, M., Liu, K. & Eksioglu, B. (2008) "A column generation approach for the split delivery vehicle routing problem", *Operations Research Letters,* Vol. 36, No. 2, pp. 265-270.

Lee, C., Epelman, M.A., White III, C.C. & Bozer, Y.A. (2006) "A shortest path approach to the multiple-vehicle routing problem with split pick-ups", *Transportation Research Part B: Methodological,* Vol. 40, No. 4, pp. 265-284.

Mitra, S. (2008) "A parallel clustering technique for the vehicle routing problem with split deliveries and pickups", *The Journal of the Operational Research Society,* Vol. 59, No. 11, pp. 1532-1546.

Mitra, S. (2005) "An Algorithm for the Generalized Vehicle Routing Problem with Backhauling", *Asia-Pacific Journal of Operational Research,* Vol. 22, No. 2, pp. 153-169.

Mizrak Özfirat, P. & Özkarahan, I. (2007) "Comparison of Non-Split and Split Delivery Strategies for the Heterogeneous Vehicle Routing Problem", *Journal of Industrial Engineering,* Vol. 14, No. 4, pp. 2-13.

Mota, E., Campos, V. & Corberán, Á. (2007) "A New Metaheuristic for the Vehicle Routing Problem with Split Demands", Anonymous *Evolutionary Computation in Combinatorial Optimization,* Lecture Notes in Computer Science, Vol. 4446/2007, pp. 121-129, Springer Berlin / Heidelberg, Germany.

Mullaseril, P.A., Dror, M. & Leung, J. (1997) "Split-delivery routeing heuristics in livestock feed distribution", *The Journal of the Operational Research Society,* Vol. 48, No. 2, pp. 107-116.

Nowak, M.A. (2005) *The pickup and delivery problem with split loads,* Ph.D. Thesis, Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA.

Nowak, M.A., Ergun, O. & White III, C.C. (2008) "Pickup and Delivery with Split Loads", *Transportation Science,* Vol. 42, No. 1, pp. 32-43.

Salkin, H.M. & Mathur, K., (1989) *Foundations of integer programming,* Elsevier Science Ltd (September 1989), New York, NY : North-Holland, 776 p.

Sierksma, G. & Tijssen, G.A. (1998) "Routing helicopters for crew exchanges on off-shore locations", *Annals of Operations Research,* Vol. 76, No. 1-4, pp. 261-286.

Solomon, M.M. (1987) "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints", *Operations Research,* Vol. 35, No. 2, pp. 254-265.

Song, S.H., Lee, K.S. & Kim, G.S. (2002) "A practical approach to solving a newspaper logistics problem using a digital map", *Computers & Industrial Engineering,* Vol. 43, No. 1-2, pp. 315-330.

Tavakkoli-Moghaddam, R., Safaei, N., Kah, M.M.O. & Rabbani, M. (2007) "A New Capacitated Vehicle Routing Problem with Split Service for Minimizing Fleet Cost by Simulated Annealing", *Journal of the Franklin Institute,* Vol. 344, No. 5, pp. 406-425.

Toth, P. & Vigo, D. (2001) "The vehicle routing problem", *SIAM monographs on discrete mathematics and applications,* Philadelphia, US: Society for Industrial and Applied Mathematics, USA.

APPENDIX: The TCRAPST as an AMPL Model

```
# SETS AND PARAMETERS
# ------------------------------------------------------------------------------------
set TASK ordered;                                          # ordered set of tasks
set WORKER     ;                                           # set of resources
set TRANSF = {i in TASK, j in TASK: ord(i) < ord(j)};      # allowed transfers between
                                                           # customer facilities

param capacity {WORKER} = 8;              # capacity of resource (here 8 assumed)
param distance {TASK, TASK} >= 0;         # set-up time between tasks
param requirement {TASK};                 # capacity required by task
param workercost {WORKER};                # cost of using resource


# VARIABLES
# ------------------------------------------------------------------------------------
var Place {WORKER, TASK} binary;          # 1, if resource visits task; 0 otherwise
var Route {WORKER, TRANSF} binary;        # 1, if resource moves between two tasks; 1
otherwise
var Beginning {WORKER, TASK} binary;      # determines the first task of a resource
var End {WORKER, TASK} binary;            # determines the last task of a worker
var WorkTime {WORKER, TASK};              # capacity time of resource used in task


# OBJECTIVE FUNCTION
# ------------------------------------------------------------------------------------
minimize Worker_Time_Needed:
	sum{w in WORKER}(
		sum{(i,j) in TRANSF}Route[w,i,j]*distance[i,j]
			+ sum{f in TASK}WorkTime[w,f])*workercost[w];


# CAPACITY CONSTRAINTS
# -------------------------------------------------------------------------------------
# ALL WORK HAS TO BE DONE
subject to WorkDemand {i in TASK}:
	sum {w in WORKER} WorkTime[w,i] = requirement[i];

# WORK, TRAVELING AND SLACK CAN NOT EXCEED CAPACITY
subject to WorkSupply {w in WORKER}:
	sum {i in TASK} WorkTime[w,i]
	+ sum {(i,j) in TRANSF} Route[w,i,j] * distance[i,j] <= capacity[w];

# FOR WORK, A WORK ASSIGNMENT DECISION HAS TO BE MADE
subject to Assignment {w in WORKER, i in TASK}:
	WorkTime[w,i] <= Place[w,i]*8 ;
```

```
# INTRAPERIOD SET-UP/TRAVELING CONSTRAINTS
# -------------------------------------------------------------------------------------
# 1) MOVEMENTS CAN ONLY TAKE PLACE IF ALLOCATION DECISION HAS BEEN MADE
# 2) A RESOURCE HAS TO BE ALLOCATED TO AT LEAST ONE TASK
subject to Work_For_Traveling {w in WORKER, f in TASK}:
	Place[w,f]
	- (sum{j in TASK: ord(j)<ord(f)} Route[w,j,f] + sum{j in TASK:
ord(j)>ord(f)}Route[w,f,j]
	+ Beginning[w,f] + End[w,f])/2 = 0 ;


# IN AND OUT OF THE SYSTEM
# -------------------------------------------------------------------------------------

subject to Worker_Has_To_Come_From_Somewhere {w in WORKER} :
	sum{f in TASK} Beginning[w,f] = 1 ;

subject to Worker_Has_To_Go_Somewhere {w in WORKER} :
	sum{f in TASK} End[w,f] = 1 ;


# SUBTOUR CONSTRAINTS
# -------------------------------------------------------------------------------------
```

Pasi P. Porkka

Testing Of Different Time Capacitated Resource Allocation Models In Service Applications

Department of Business Technology

January
2010

AALTO UNIVERSITY
SCHOOL OF ECONOMICS

UNPUBLISHED WORKING PAPER

Abstract

Porkka (2009) modeled a Time Capacitated Resource Allocation Problem with Split Tasks (TCRAPST) and showed a theoretical 50% capacity time savings potential of allowing time capacitated task splitting in a service resource allocation problem where both work and traveling between tasks consume, and are measured as, capacity time. The TCRAPST allows more than one resource working in each task when a Time Capacitated Resource Allocation Problem (TCRAP) modeled in this essay requires one worker per task. Both MILP models minimize the number of resources needed. In this essay, the TCRAPST is modeled and tested for capacity time savings against the TCRAP. Test problem sets differ from each other by average task lengths and average distances between tasks.

The TCRAP is a basic routing and allocation problem and similar models are likely to be found in literature. The contribution of this essay lies on using the TCRAP as a reference model to the TCRAPST. The new test problems sets may also be useful in evaluating related models emerging in the future.

Capacity time savings varied substantially between problem sets. Even 33% savings are reported in the average number of workers needed when average task lengths were just above half of the capacity of resources, task length variation was small and the distances relatively short. Test results suggest that task splitting should be considered as an integral part in time capacitated planning models.

Key Words: service resource allocation, set-up time, time capacitated planning, scheduling, services, MILP

List of Contents

List of Figures

List of Tables

1. Introduction

In many planning and scheduling situations time is an important constraint. Time can measure the length of work in a task as well as the switching time between tasks. In production, machines with fixed locations are the resources and tasks are allocated to them. In many services, however, tasks have fixed locations and resources are allocated to tasks. For example, in house cleaning, houses have a fixed location and cleaning people move from house to house. The time of resources can be divided into three components that are working time, moving time and slack. If the time needed for performing tasks and moving between tasks is predictable, time capacitated routing and scheduling can be applied to minimize the amount of resources needed.

In Porkka (2009) it was shown that the time capacitated splitting of tasks to be performed by more than one resource can generate more efficient plans than when splits are not allowed. In this essay 50 simulated problems are solved to study the effects of splits in real planning situations. Problems are divided into 10 sets with task lengths and distances being generated from different probability distributions. Each problem includes 12 tasks and the purpose of the problem sets is to simulate a cluster of tasks.

Problems are first solved using the Time Capacitated Resource Allocation Problem with Split Tasks (TCRAPST) model formulated in Porkka (2009) and then by the Time Capacitated Resource Allocation Problem (TCRAP) that solves the same problems but without splits. The TCRAP is a basic routing and allocation problem and similar models are likely to be found in literature. For this essay, the TCRAP was modeled to be a reference model to the TCRAPST when testing with simulated problems the savings effects of time capacitated task splitting. The use of the TCRAP for that purpose is new because, according to author's knowledge, the TCRAPST was first modeled in Porkka (2009) and this essay is the first research where its performance is compared with a reference model.

Most test problems were solved to optimality by the TCRAP but with the TCRAPST they could be solved to optimality in some cases only. Near optimal solutions were seen sufficient to demonstrate the savings generated by split tasks and to describe some interesting characteristics of solutions.

In test problems, the resource capacity is set to be the same as the length of the planning period which makes the analysis of solutions easier. In examples the length of a planning period and the capacities of each resource are 8 hours which can be interpreted as the length of a working day. An easy example for a reader to keep in mind is a set of cleaning tasks in different locations and a set of workers that have to be allocated to do those tasks. The main cost in test problems is the resource time which is different for each resource. In this way, the utilization of least expensive resources gets maximized and the utilization of most expensive resources gets minimized.

Chapter 2 gives a short literature research on Split Delivery Vehicle Routing Problem (SDVRP). Chapter 3 restates the TCRAPST from Porkka (2009) and presents the TCRAP formulation. In Chapter 4 gives an example of a time capacitated service allocation problem with 25 tasks solved by the TCRAPST and the TCRAP. The typical features of solutions are described and discussed. Chapter 5 describes the problem sets and test results. Chapter 6 concludes the essay.

2. Literature Research

The TCRAPST is closely related to Vehicle Routing Problem (VRP) and, especially, to its variant Split Delivery Vehicle Routing Problem (SDVRP). The VRP concerns the distribution of goods between depots and final users. Customers and typically one depot form a network usually modeled as a graph which can be either directed or non-directed. Typically transportation capacity or route length is limited leading to a situation when all customers can not be served by one route and one vehicle only. Other constraints can include periods of the day (time windows) during which customers have to be served, unloading or loading times, vehicle type, different priorities, and penalties associated with partial or total lack of service associated with customers. Routes can include deliveries, pick-ups or both. The objective is to minimize transportation costs that consist of the number of vehicles needed and actual traveling costs typically consisting of the total distance traveled. Toth and Vigo (2001) wrote a comprehensive book on Vehicle Routing Problem models and algorithms to solve them.

A VRP problem allowing a client to be served using more than one vehicle is a SDVRP. The SDVRP is a relaxation of the classical VRP, but it still remains NP-hard. The SDVRP is quite similar to the problem of allocation of moving resources to services. In the SDVRP a customer requiring products can be served by one or more vehicles whereas in service resource allocation a task can be completed by one or more service resources. In both cases routing and split decisions are required and the objective is to minimize resource costs.

The time spent at customer’s site, as products are picked up or delivered, could be considered as a constraint on vehicle routing too. In that case, the delivery splits could be based on the time that it takes to pick up and to unload products to and from a vehicle.

Sometimes both the delivery amount and service time may be constraints. For example, pumping oil from or into big tankers can take a long time and may not always be completed in one dock because the dock may need to be freed for another tanker loading or unloading another type of oil. One tanker may be able to satisfy loading or unloading requirements only partially before it has to leave. Loading and unloading of oil in tankers takes a long time and sometimes time at a dock can become a constraint.

The splitting of deliveries is not exactly the same as the splitting of time capacitated tasks, but they both have the same kind of savings potential. Therefore, understanding split deliveries in vehicle routing creates intuition of using split tasks in time capacitated service resource allocation. In vehicle routing, allowing split deliveries generates savings in both the total distance traveled and in the number of vehicles to be used.

Dror and Trudeau (1987) introduced the concept of Split Delivery Routing. Dror and Trudeau (1989) presented a Split Delivery Vehicle Routing Problem (SDVRP), developed a solution scheme and demonstrated the potential for cost savings through split deliveries by using generated problems. Archetti et al. (2008) and Porkka (2009) present comprehensive literature reviews on Split Delivery Vehicle Routing Problem. According to Porkka, in SDVRP literature the time capacitated modeling with explicit consideration on time capacitated task splitting had not been studied in literature before.

3. TCRAPST and TCRAP Model Formulations

3.1. TCRAPST Formulation

For reference and comparison, the formulation of the Time Capacitated Resource Allocation Problem with Split Tasks (TCRAPST) from Porkka (2009) is restated below.

Notations

Sets

P	= set of resources	
I	= ordered set of tasks	

Parameters

$D_{i,j} = D_{j,i}$	= set-up time between tasks i and j,	$i \in I, j \in I$
R_i	= capacity required by task i,	$i \in I$
C_p	= capacity of resource p,	$p \in P$
K_p	= cost of using resource p,	$p \in P$
M	= big number	

Binary variables

$s_{p,i}$	= 1, if resource p visits task i; 0 otherwise,	$p \in P, i \in I$
$d_{p,i,j} = d_{p,j,i}$	= 1, if resource p moves between tasks i and j; 0 otherwise,	$p \in P, i \in I, j \in I$
$b_{p,i}$	= 1, if resource p starts the period in task i; 0 otherwise,	$p \in P, i \in I$
$e_{p,i}$	= 1, if resource p ends the period in task i; 0 otherwise,	$p \in P, i \in I$

Continuous variables

$w_{p,i}$	= capacity time (or work) of resource p used in task i,	$p \in P, i \in I$
h_p	= slack time of resource p,	$p \in P$.

Objective function

$$\min \sum_{p\in P}\sum_{i\in I}\sum_{\substack{j\in I\\ i<j}} d_{p,i,j} D_{i,j} K_p + \sum_{p\in P}\sum_{i\in I} w_{p,i} K_p \tag{1}$$

The objective function attempts to minimize the number of active resources and the total set-up time simultaneously. The most important objective is to minimize the number of active resources. An active resource here refers to a resource that is used for work.

The TCRAPST allocates all resources to at least one task. However, an allocation to just one task does not require either traveling or work. So, a resource allocated to just one task can either be an active or a inactive resource.

When all resources are allocated, but all allocated resources do not necessarily either work or travel, the cost of using resources for work ($w_{p,i}K_p$) or traveling with different K_p for each resource minimizes the number of active resources needed to perform tasks. We can, for example, use costs $K_{p+1} = 10K_p$ which uses the cheapest resources first and forces more expensive resources on paths with no traveling or allocated work. Simultaneously, the slack of the most expensive activated resource is maximized.

The constraints of this model can be grouped into capacity constraints and set-up constraints. Capacity constraints regulate the distribution of resource capacity time between work, set-ups and slack. Set-up constraints define the possible routes that can exist for each resource between tasks.

Capacity constraints

$$\sum_{p \in P} w_{p,i} = R_i \qquad , \forall\, i. \qquad (2)$$

$$\sum_{i \in I} \left(w_{p,i} + h_p \right) + \sum_{i \in I} \sum_{\substack{j \in I \\ i<j}} d_{p,i,j} D_{i,j} = C_p \qquad , \forall\, p. \qquad (3)$$

$$w_{p,i} \le s_{p,i} M \qquad , \forall\, p \text{ and } i. \qquad (4)$$

Constraint (2) requires that all work in tasks gets done. According to constraint (3) time spent on work, set-ups, and slack has to equal the capacity of a resource. Because a resource has to be physically available to do its task, in constraint (4) $s_{p,i}$ is 1 for each resource p working in a task i during a period and 0 otherwise. This formulation allows a resource to be routed via one task only and without any working on its path. Resources without any allocated work can be considered as inactivated, and thus unneeded, resources.

Set-up constraints

$$\sum_{i \in I} b_{p,i} = 1 \qquad , \forall\, p. \qquad (5)$$

$$\sum_{i \in I} e_{p,i} = 1 \qquad , \forall\, p. \qquad (6)$$

$$s_{p,i} - \frac{\left(b_{p,i} + e_{p,i} \right)}{2} - \frac{\left(\sum_{\substack{j \in I \\ i<j}} d_{p,i,j} + \sum_{\substack{j \in I \\ j<i}} d_{p,j,i} \right)}{2} = 0 \qquad , \forall\, p \text{ and } i. \qquad (7)$$

$$w_{p,i}, h_p \ge 0 \qquad , \forall\, p, \text{ and } i,$$

$$b_{p,i}, d_{p,i,j}, e_{p,i}, s_{p,i} \in \{0,1\} \qquad , \forall\, p, i, \text{ and } j.$$

Constraints (5) and (6) forces there to be one first and one last task for each resource p. Constraint (7) lets set-ups or movements take place only to and from tasks i with working assignment $s_{p,i} = 1$. Constraint (7) requires at least one assigned task for each resource. To find a realistic solution to the TCRAPST, we typically have to generate additional subtour constraints during the solution process. More detailed description of the TCRAPS can be found in Porkka (2009).

3.2. TCRAP Formulation

The TCRAP is a modification of the TCRAPST. The TCRAP requires each task to be completed by one resource only. However, the formulation allows more than one resource to be routed via tasks that do not require any work to do. This feature can be used to route inactive resources. Inactive workers may exist because initially the actually number of needed resources is not know. Therefore, to guarantee the feasibility of a problem it is wise to start with extra capacity and let the model determine the real capacity needed. To solve the TCRAP using this strategy, one needs a task with a zero requirement to allocate the potentially inactive resources. As inactive resources are routed through that dummy task only, they follow a one-task-path where their capacity is neither used for either traveling nor working. As no one of the resources allocated to that dummy task is working, task splitting does not need to be done within that task.

If the distance from to the dummy task ($\delta \in I$ and $R_\delta = 0$) to other tasks is long, only active workers get allocated to real tasks as all inactive workers are routed via the dummy task only. The dummy task without requirements can be generated simultaneously with the normal tasks, i.e. $i \in I \Rightarrow i \in \{I, \delta\} = I'$.

The TCRAP formulation and differences with regard to the TCRAPST are stated below:

Objective function

$$\min \quad \sum_{p \in P} \sum_{i \in I'} \sum_{\substack{j \in I' \\ i<j}} d_{p,i,j} D_{i,j} K_p + \sum_{p \in P} \sum_{i \in I'} s_{p,i} R_i K_p \qquad (8)$$

Objective function (8) states that the whole cost of work in a task is charged to one resource. It minimizes the same thing as the objective function (1) in the TCRAPST formulation.

Capacity constraints

$$\sum_{p \in P} s_{p,i} = 1 \qquad , \forall\, i \neq \delta \qquad (9)$$

$$\sum_{i \in I} \left(w_{p,i} + h_p \right) + \sum_{i \in I} \sum_{\substack{j \in I \\ i<j}} d_{p,i,j} D_{i,j} = C_p \qquad , \forall\, p. \qquad (10)$$

$$w_{p,i} = s_{p,i} R_i \qquad , \forall\, p \text{ and } i. \qquad (11)$$

Capacity constraint (9) states that all real tasks have to be visited. Constraint (10) is the same as (3) in the TCRAPST. Constraint (11) requires all work in a task to be performed by one resource. This constraint also allows many workers to be allocated to a dummy task. Set-up and subtour constraints in the TCRAP are the same as in the TCRAPST.

4. Example Problem Solved by the TCRAPST and the TCRAP

In the TCRAPST and the TCRAP both capacity and tasks are measured as time. Resource capacity is used for work, travelling or staying idle. A time capacitated resource can be a person, an animal, a machine, a vehicle, a transportable stage for an activity etc. The key is that time required by different tasks and the production rate of different resources can be estimated. (Sometimes, when only presence is needed, as in many security services, we do not even need the production rate. Then both requirements and the provided service is measured as time, but presence time is constrained by time windows not included in the TCRAPST or the TCRAP.) Resources are allocated to tasks and one resource may work in more than one task. The TCRAP model requires each resource allocated to a task to complete that task. The TCRAPST model allows the work in a task to be split between several resources. The objective in the TCRAPST and the TCRAP is to minimize the number of active resources needed. Simultaneously, the slacks of the last, and most expensive, active resources are maximized.

The TCRAP and the TCRAPST could, for example, answer a question:

> *How to allocate 25 cleaning people within an 8 hour work shift to 25 tasks with different durations so that the number of people actually needed is minimized?*

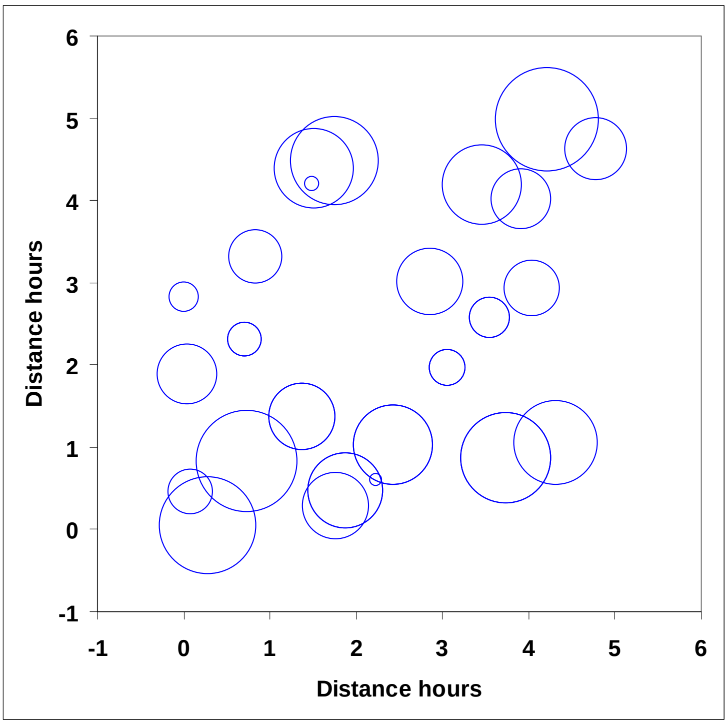

Figure 1. Tasks scattered on a “map”

Figure 1 illustrates the problem with the centers of circles corresponding to task locations, the diameters of circles to task lengths, and euclidean distances between the centers of circles to traveling times between tasks. Distances between the centers of circles correspond to the distance hour scale of the axes. The diameters of circles do not directly relate to axes. A big circle is a long task and a small circle is a short task. Initially 25 workers with 8 hours of capacity are made available. The TCRAPST and the TCRAP solutions give estimates for how many workers are actually needed.

For this example, the lengths of tasks are randomly generated from the normal distribution N[1,8]. Horizontal and vertical coordinates are randomly generated from the normal distribution N[0,5]. No minimum working time in tasks is set for split tasks. The simulated task lengths and coordinates are listed in Table 1.

The cost of time spent on traveling was the same cost for each worker, but, for each worker, the working time in tasks was given a different cost so that the working time of a worker, or the time spend in tasks, increases as worker's ordinal number increases, i.e. worker number one is the cheapest to use and worker number 25 is the most expensive. As working time in tasks varies, but the cost of traveling is the same for each worker, the solver tries to find solutions where least expensive workers are used and the use of most expensive workers is avoided.

Table 1. Task lengths and coordinates for the example problem

Task number	1	2	3	4	5	6	7	8	9	10	11	12	13
Horizontal	0.01	4.04	1.75	3.73	3.55	0.07	0.74	2.23	0.04	2.86	0.83	1.76	3.92
Vertical	2.82	2.93	4.48	0.87	2.57	0.46	0.83	0.60	1.89	3.01	3.32	0.29	4.01
Task length	2.35	4.36	6.76	7.01	3.13	3.55	7.92	1.03	4.72	5.25	4.16	5.25	4.64

Task number	14	15	16	17	18	19	20	21	22	23	24	25
Horizontal	1.51	4.78	0.71	4.31	4.22	3.06	1.49	1.88	0.28	1.38	3.46	2.42
Vertical	4.38	4.63	2.31	1.05	4.98	1.96	4.20	0.46	0.04	1.36	4.19	1.03
Task length	6.09	4.78	2.65	6.46	8.00	2.86	1.17	5.74	7.43	5.12	6.09	6.21

Distances (or set-up time) between tasks are calculated as a euclidean distance between task coordinates. Based on task lengths and distances between tasks, each cleaning person can, on average, be estimated to work in less than 2 tasks.

Figure 2 illustrates solutions to the example generated by the TCRAP and the TCRAPST. The TCRAP requires that a task should be completed by one worker only. This leaves a considerable amount of slack for almost all workers. The TCRAPST allows tasks to be shared. In the TCRAP solution workers move less between tasks than in the TCRAPST solution. There are only 5 TCRAP movements as there are 17 movements in the TCRAPST solution.

Due to problem complexity, these solutions are only indicative, not optimal. This can be also seen in Table 2 where working time of workers does not always decrease as the worker number increases. In an optimal solution, the utilization of cheaper workers in tasks should be higher than the utilization of more expensive workers. Still, even these non-optimal solutions are useful because they illustrate the typical solution features of each of two models.

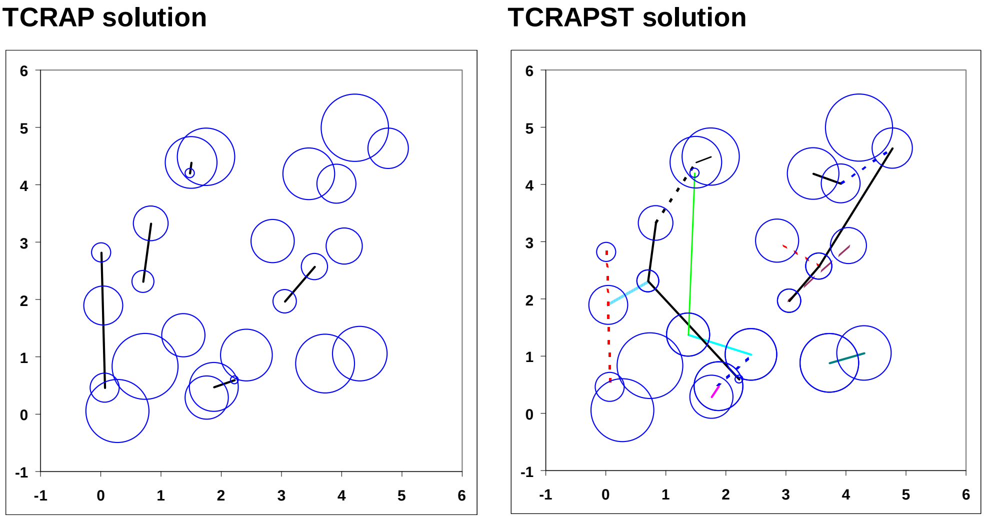

Figure 2. Solutions generated by the TCRAP and the TCRAPST

As the solutions in Table 2 indicate, fewer workers are needed if task splitting is allowed, even, when more time is spent on traveling. In the TCRAP solution, 77% of active workers' capacity is spent on working, 3% is spent on traveling and 20% is spent on doing nothing. In the TCRAPST solution, 81% of used workers' capacity is spent on working, 13% is spent on traveling and 6% is spent on doing nothing. Workers that were not allocated either work or traveling were not considered as active workers. Typically, the active worker with the highest ordinal number is the most expensive one and has bigger slack than other active workers.

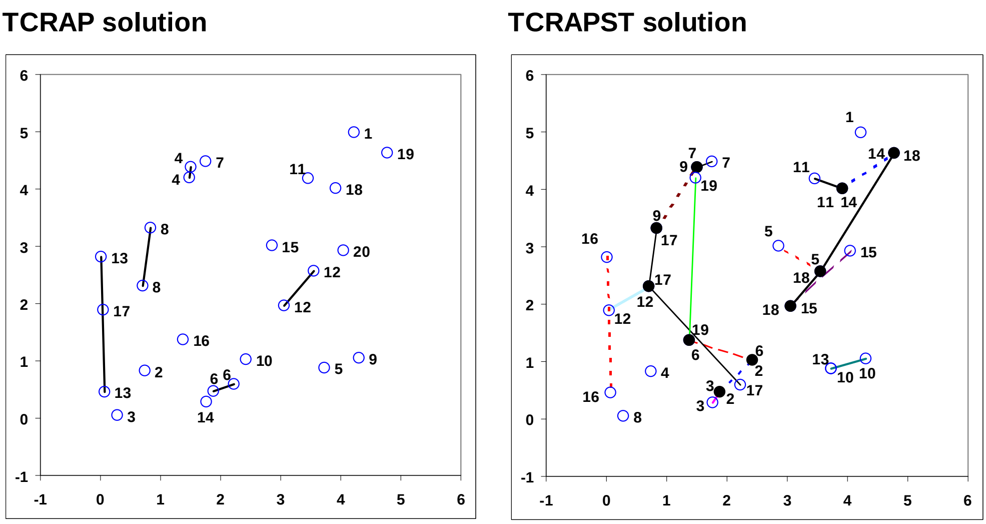

Figure 3. Workers allocated to tasks in solutions generated by the TCRAP and the TCRAPST

In Figure 3, only the locations of tasks are shown but the length of tasks is ignored. In the TCRAPST solution, shared tasks are colored with black. The numbers indicate the ordinal number of workers working in each task. In the TCRAP solution, only 5 workers move between tasks when in the TCRAPST solution workers change place 17 times during their work shift. The TCRAP solution in Table 3 requires 20 workers to complete 25 tasks when the TCRAPST solution in Table 5 needs only 19 workers by sharing 10 tasks between more than one worker. The TCRAPST solution in Table 3 shows very short working times indicating the need for introducing minimum working time constraints in practical applications.

Table 2. Working times, traveling times, utilization for tasks and slack

TCRAP solution

Worker	Working time	Traveling time	Working + Traveling	Utilization for tasks	Slack
1	8.00		8.00	100 %	
2	7.92		7.92	99 %	1 %
3	7.43		7.43	93 %	7 %
4	7.26	0.45	7.71	91 %	4 %
5	7.01		7.01	88 %	12 %
6	6.77	0.70	7.47	85 %	7 %
7	6.76		6.76	85 %	16 %
8	6.81	1.07	7.88	85 %	2 %
9	6.46		6.46	81 %	19 %
10	6.21		6.21	78 %	22 %
11	6.09		6.09	76 %	24 %
12	5.99	1.05	7.04	75 %	12 %
13	5.90	1.56	7.46	74 %	7 %
14	5.25		5.25	66 %	34 %
15	5.25		5.25	66 %	34 %
16	5.12		5.12	64 %	36 %
17	4.72		4.72	59 %	41 %
18	4.64		4.64	58 %	42 %
19	4.78		4.78	60 %	40 %
20	4.36		4.36	55 %	46 %

Average capacity used in tasks	77 %
Average capacity spent on traveling	3 %
Average slack per worker	20 %

TCRAPST solution

Worker	Working time	Traveling time	Working + Traveling	Utilization for tasks	Slack
1	8.00		8.00	100 %	
2	6.94	1.06	8.00	87 %	
3	7.45	0.55	8.00	93 %	
4	7.92		7.92	99 %	1 %
5	6.93	1.07	8.00	87 %	
6	6.82	1.18	8.00	85 %	
7	7.41	0.59	8.00	93 %	
8	7.43		7.43	93 %	7 %
9	6.67	1.33	8.00	83 %	
10	7.13	0.87	8.00	89 %	
11	7.20	0.80	8.00	90 %	
12	6.87	1.05	7.92	86 %	1 %
13	6.34		6.34	79 %	21 %
14	6.78	1.22	8.00	85 %	
15	6.60	1.40	8.00	83 %	
16	5.90	1.56	7.46	74 %	7 %
17	4.46	2.87	7.33	56 %	8 %
18	3.60	2.87	6.47	45 %	19 %
19	2.28	1.72	4.00	28 %	50 %

Average capacity used in tasks	81 %
Average capacity spent on traveling	13 %
Average slack per worker	6 %

Table 3. Tasks allocated per worker by the TCRAP

Worker	1	2	3	4		5	6		7
Task number	18	7	22	14	20	4	8	21	3
Working time per task	8,00	7,92	7,43	6,09	1,17	7,01	1,03	5,74	6,76
Total working time	8,00	7,92	7,43	7,26		7,01	6,77		6,76

Worker	8		9	10	11	12		13	
Task number	11	16	17	25	24	5	19	1	6
Working time per task	4,16	2,65	6,46	6,21	6,09	3,13	2,86	2,35	3,55
Total working time	6,81		6,46	6,21	6,09	5,99		5,90	

Worker	14	15	16	17	18	19	20
Task number	12	10	23	9	13	15	2
Working time per task	5,25	5,25	5,12	4,72	4,64	4,78	4,36
Total working time	5,25	5,25	5,12	4,72	4,64	4,78	4,36

Table 4. Tasks allocated per worker by the TCRAPST

Worker	1	2		3		4	5		6	
Task number	18	21	25	12	21	7	5	10	23	25
Working time per task	8,00	3,54	3,40	5,25	2,20	7,92	1,68	5,25	4,01	2,81
Total working time	8,00	6,94		7,45		7,92	6,93		6,82	

Worker	7		8	9		10		11	
Task number	3	14	22	11	14	4	17	13	24
Working time per task	6,76	0,65	7,43	1,23	5,44	0,67	6,46	1,11	6,09
Total working time	7,41		7,43	6,67		7,13		7,20	

Worker	12		13	14		15		16	
Task number	9	16	4	13	15	2	19	1	6
Working time per task	4,72	2,15	6,34	3,53	3,25	4,36	2,24	2,35	3,55
Total working time	6,87		6,34	6,78		6,60		5,90	

Worker	17			18			19	
Task number	8	11	16	5	15	19	20	23
Working time per task	1,03	2,93	0,5	1,45	1,53	0,62	1,17	1,11
Total working time	4,46			3,6			2,28	

Table 5. Workers allocated per task

TCRAPST solution

Task number	1	2	3	4		5		6	7	8
Worker	16	15	7	10	13	5	18	16	4	17
Working time per task	2.35	4.36	6.76	0.67	6.34	1.68	1.45	3.55	7.92	1.03
Total work in task	2.35	4.36	6.76	7.01		3.13		3.55	7.92	1.03

Task number	9	10	11		12	13		14	
Worker	12	5	9	17	3	11	14	7	9
Working time per task	4.72	5.25	1.23	2.93	5.25	1.11	3.53	0.65	5.44
Total work in task	4.72	5.25	4.16		5.25	4.64		6.09	

Task number	15		16		17	18	19		20
Worker	14	18	12	17	10	1	15	18	19
Working time per task	3.25	1.53	2.15	0.5	6.46	8.00	2.24	0.62	1.17
Total work in task	4.78		2.65		6.46	8.00	2.86		1.17

Task number	21		22	23		24	25	
Worker	2	3	8	6	19	11	2	6
Working time per task	3.54	2.20	7.43	4.01	1.11	6.09	3.40	2.81
Total work in task	5.74		7.43	5.12		6.09	6.21	

In this example, the TCRAPST needed one worker less than the TCRAPS making a 5% saving in work force. The test problems in the following chapters show that, depending on task lengths and task distances, the savings from splitting tasks can be bigger or smaller than 5%.

5. Testing the Savings Potential with Simulated Problems

5.1. Test Plan

To demonstrate the savings potential of task splitting, ten sets of test problems were generated and solved by the TCRAP and the TCRAPST. The main purpose of tests was to compare the number of resources needed in TCRAP and TCRAPST solutions. Other solution features, such as (1) total work time + traveling, (2) last worker work + traveling, (3) last worker slack, and (4) the number of splits were also compared. When analysing the test solutions words resource and worker are used interchangeably with the same meaning.

I the test problems, the 8-hours planning period, 8 hours of capacity of each resource, the less than 8 hours long tasks and the less than 1 hour distances between tasks simulate an 8-hour working day. Locations of tasks mimic a city center or a suburb where the average minimum distance from each task to the closest neighboring task takes about 0,2 – 1 hours to travel. Task lengths and locations were generated from uniform distributions. One resource with 8 hours capacity was made available per each task.

Each problem consisted of 12 tasks only because bigger TCRAP problems could not be solved to optimality with our multiprocessor desktop computer (4*Intel(R)Xeon(R) CPU E5420@2.50GHZ and 2.49GHZ, 3.00 GB of RAM) and CPLEX 9.1. The branch-and-bound tree (CPLEX option `treememlim)` was limited to 500 MB to get TCRAPST solutions in reasonable solution time. Even though most of the TCRAPST solutions generated in this way were not optimal, they still were able to demonstrate the substantial savings potential of time capacitated task splitting.

5.2. Objective Function

Savings potential of the TCRAPST in comparison to the TCRAP was tested by using objective functions (1) and (8). Three other objective function formulations were also experimented with but objective functions (1) and (8) were more efficient in "pushing" inactive resources out.

$$\min \sum_{p\in P}\sum_{i\in I}\sum_{\substack{j\in I\\ i<j}} d_{p,i,j}D_{i,j} + \sum_{p\in P}\sum_{i\in I}\left(w_{p,i} - s_{p,\delta}\right)K_p \tag{12}$$

$$\min \sum_{p\in P}\sum_{i\in I}\sum_{\substack{j\in I\\ i<j}} d_{p,i,j}D_{i,j} + \sum_{p\in P}\sum_{i\in I}\left(w_{p,i} - s_{p,\delta}\right)K_p + \sum_{p\in P}\sum_{\substack{i\in I\\ i\neq\delta}}\left(Qb_{p,i}\right)K_p \tag{13}$$

$$\min \quad \sum_{p\in P}\sum_{i\in I}\sum_{\substack{j\in I\\ i<j}} d_{p,i,j}D_{i,j}K_p + \sum_{p\in P}\sum_{i\in I} w_{p,i}K_p \tag{14}$$

Objective function (12) in TCRAPST reached the same solutions as objective function (1), but solution time was longer because this formulation required more subtour constraints than objective function (1). The purpose of the negative allocation cost $s_{p,\delta}K_p$ ($\delta \in I$ and $R_\delta = 0$) in objective function (12) was to intensify the search for solutions where the most expensive resources are not used for work.

Objective function (13) used the same negative allocation cost $s_{p,\delta}K_p$ as objective function (12) but also added a cost $Qb_{p,i}K_p$ to all paths not starting at a non-requirement task δ ($\delta \in I$ and $R_\delta = 0$). (Q has to be less than M because otherwise all paths would start from δ.) This formulation was also aimed to intensify the search for solutions where the most expensive resources are not used for work. However, the approach did not work because it continued creating new subtour constraints until all memory on the desktop computer was used.

An objective function where both traveling time and working time had cost was more efficient in getting compressed solutions than objective function (14) where only working time was charged. In addition to that, the latter objective function needed many more subtour prevention constraints than the first one and the generation of those subtour prevention constraints made the solution process much longer than with the first objective function.

We could also use an objective function that minimizes the number of resources needed without any incremental costing of resources. After finding out the minimum number of workers we could then fix the number of workers to that minimum and then apply an objective function that squeezes out the maximum slack from the last resource that is used.

The parametrization of objective function affects the efficiency of the solution process. Because the goal was to minimize the number of resources, usage of incremental costing of resources was found useful. By using incremental costs, sufficient amount of resources can be made available to guarantee the feasibility of problems. The more difference there is between the costs of different resources the more *compressed* are the solutions. Table 6 shows three different resource costing schemes and Table 7 shows three examples of solutions that the TCRAPST might generate. In Table 7, more compression thus means a solution with fewer activated resources needed.

Table 6. Different resource costs

	Resource Cost		
Resource	$K_{n+1} = 2K_n$	$K_{n+1} = 10K_n$	$K_{n+1} = 100K_n$
1	1	1	1
2	2	10	100
3	4	100	10000
⋮	⋮	⋮	⋮
12	2048	1E+11	1E+22

Table 7. Different levels of compression in solutions depend on resource costs

	Working + Traveling Hours per Resource		
Resource	**Resource cost** $K_{n+1} = 2K_n$	**Resource cost** $K_{n+1} = 10K_n$	**Resource cost** $K_{n+1} = 100K_n$
1	8.00	8.00	8.00
2	8.00	8.00	8.00
3	8.00	8.00	8.00
4	7.40	8.00	8.00
5	8.00	8.00	8.00
6	6.70	7.50	7.87
7	3.00	2,00	
8	0.25		

Low Compression **High Compression**

So, why not to use the highest resource cost differences? The reason for that is the scaling problems of CPLEX emerging from a very large variety of resource costs. As parameters have very different dimensions, CPLEX has difficulties in finding feasible solutions and soon gets completely stuck.

Resource cost scheme $K_{n+1} = 10K_n$ was used in test problems. The scaling for CPLEX was made easier by setting the first resource cost to 0.000001. In that way, we got for 12 resources a cost scale that reached from 0.000001 to 100 000. Scheme $K_{n+1} = 2K_n$ was rejected because it produced too low compression. Scheme $K_{n+1} = 100K_n$, on the other hand, worked well with some problem sets but jammed the solver with most of the problem sets. Therefore it was rejected, too.

As $b_{p,i}$ and $e_{p,i}$ do not incur costs, visiting a task in the TCRAPST model without working does not cost anything. So, all workers visit at least one task, but only part of workers are allocated to work.

In the TCRAP however, a worker allocated to a task has to do the whole task. Therefore, to route workers not actually working, we need an additional dummy task ($\delta \in I$) with no requirement. Visiting such a task does not cost anything. As we also set the distance between the dummy task and other tasks very long, allocates the TCRAP all inactive workers to that dummy task.

The same problems have to be stated slightly differently to the TCRAP and the TCRAPST. As the generation of just 12 tasks was enough for the TCRAPST, 13 tasks were needed to the TCRAP one of those tasks being a dummy task.

5.3. Subtour Constraints

With each problem, after running the solver for some time solutions were tested for subtours and subtour constraints were added if needed. After adding subtour constraints, the solution process of the now subtour constrained problem was started from the beginning. If new subtours did not exist in a test, the solver was allowed to continue from the current solution. The incremental addition of subtours was necessary because even with small problems the initial creation of all possible subtour constraints would make problems too constrained for the CPLEX.

5.4. Problem Sets

Table 8 lists the different problem settings used to demonstrate that savings from task splitting highly depend on task lengths and the relationship between task lengths and distances. 5 problems were generated for each of the 10 test series. Then all 50 different problems were solved by the TCRAP and the TCRAPST models. Finally, the solutions were compared to better understand the different solutions generated by two different models.

Table 8. Task lengths and locations

Test series number	Average task length	Variation in task length	Distances		Task length distribution	Horizontal coordinate distribution	Vertical coordinate distribution
1	Short	Small	Short		U[0,5 , 1,5]	U[0 , 1]	U[0 , 1]
2	Short	Small	Long		U[0,5 , 1,5]	U[0 , 5]	U[0 , 5]
3	1/2*day	Big	Short		U[0,5 , 7,5]	U[0 , 1]	U[0 , 1]
4	1/2*day	Big	Long		U[0,5 , 7,5]	U[0 , 5]	U[0 , 5]
5	Long	Small	Short		U[6,5 , 7,5]	U[0 , 1]	U[0 , 1]
6	Long	Small	Long		U[6,5 , 7,5]	U[0 , 5]	U[0 , 5]
7	>1/2*day	Small	Short		U[4,0 , 5,0]	U[0 , 1]	U[0 , 1]
8	>1/2*day	Small	Long		U[4,0 , 5,0]	U[0 , 5]	U[0 , 5]
9	<1/2*day	Small	Short		U[3,0 , 4,0]	U[0 , 1]	U[0 , 1]
10	<1/2*day	Small	Long		U[3,0 , 4,0]	U[0 , 5]	U[0 , 5]

5.5. Test Results

Table 9 presents a summary of test problem solutions. Numerical values except for the counting of optimal solutions are averages of 5 problems. As problems were solved, CPLEX option `treememlim` was set to 1500 for the TCRAP and 500 for the TCRAPST. If CPLEX option `treememlim` for the TCRAPST had been set higher, higher compression in solutions could possibly have been achieved. On the other hand, the difference between TCRAPST solutions with `treememlim` 100 and `treememlim` 500 was so small that substantial improvements in TCRAPST solutions with `treememlim` bigger than 500 are not likely.

The column *Objective Function* in Table 9 presents objective function values but they are not really relevant because resource costs were artificial and the big cost differences between resources were only generated to manipulate the solver to find the minimum number of workers needed.

The column *Number Of Optimal Solutions* in Table 9 indicates that it was very difficult to find an optimal solution to the TCRAPST problems. However, for the TCRAP problems optimal solutions were looked for to have a solid base for comparison. There was only one TCRAP problem where CPLEX could only find an objective function value 0.000000005015576% from the optimum (Solution: 279 069.6024**28**; Optimum: 279 069.6024**14**) the gap being so small that the problem can be considered as solved in practice.

The column *Number Of Workers* in Table 9, Figure 4 and Figure 5 show the average number of workers used by the two models. In 4 problem sets of 10, the TCRAP and the TCRAPST used an equal number of workers. In 6 of 10 problem sets the TCRAPST was able to find solutions with fewer workers than the TCRAP.

The biggest difference in the number of activated resources needed can be seen in problem sets 7 and 8 where the TCRAP allocates only one task per worker leaving much slack as the TCRAPST through task splitting can leave most expensive resources unused.

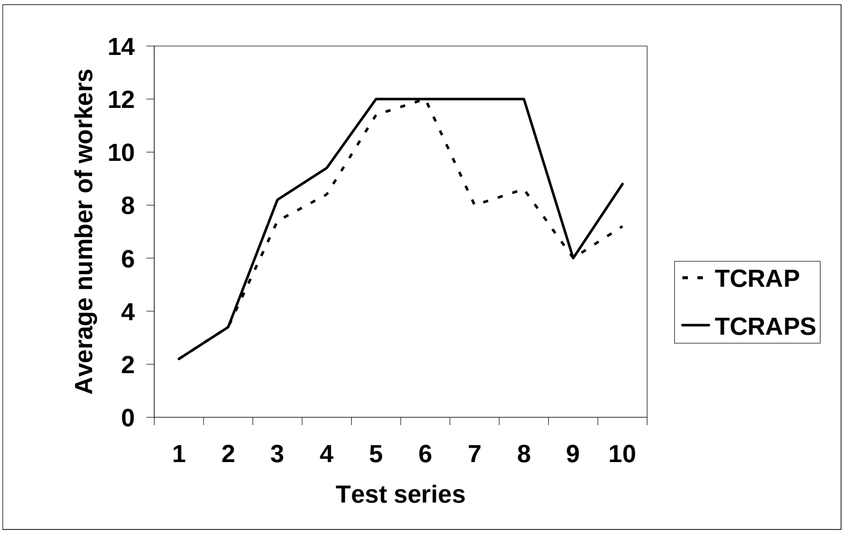

Figure 4. Average number of workers needed in test problems

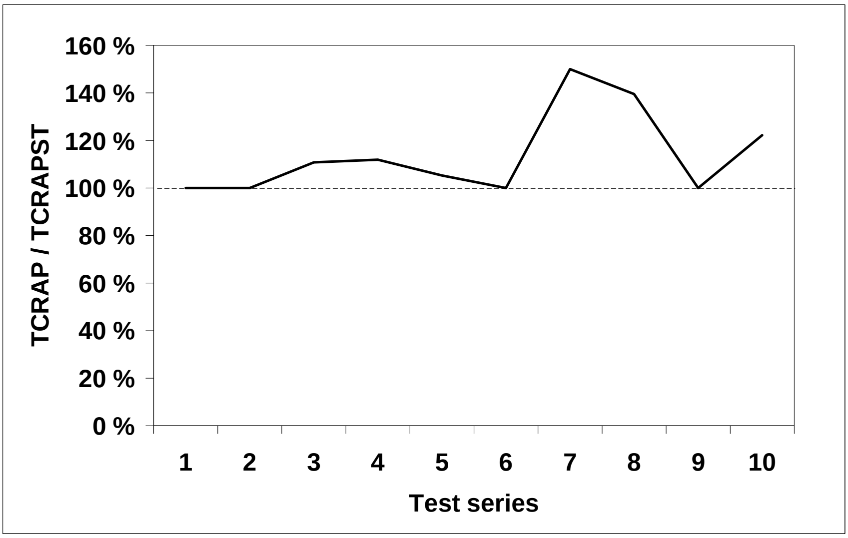

Figure 5. Average number of workers needed in test problems

Table 9. Summary of test problem solutions

Set	Task Length	Co-ordinates	Model	Objective Function	Number Of Optimal Solutions	Number Of Workers	Total Work Time + Traveling	Last Worker Work + Traveling	Total Slack	Last Worker Slack	Else Than Last Worker Slack	Number Of splits
1	U[0.5, 1.5]	U[0.0, 1.0]	TCRAPST	0	5	2,2	14,89	5,32	2,71	0,03	2,68	0,8
			TCRAP	0	5	2,2	14,83	5,48	2,77	0,25	2,52	
			TCRAP/TCRAPST	116 %	0	100 %	100 %	103 %	102 %	735 %	94 %	
2	U[0.5, 1.5]	U[0.0, 5.0]	TCRAPST	0	5	3,4	22,28	3,80	4,92	0,72	4,20	0,2
			TCRAP	0	5	3,4	22,01	3,86	5,19	1,05	4,14	
			TCRAP/TCRAPST	103 %	0	100 %	99 %	101 %	106 %	145 %	99 %	
3	U[0.5, 7.5]	U[0.0, 1.0]	TCRAPST	16	0	7,4	53,65	2,56	5,55	0,11	5,44	5,6
			TCRAP	822	5	8,2	52,66	3,49	12,94	8,43	4,51	
			TCRAP/TCRAPST	5159 %	0	111 %	98 %	136 %	233 %	7527 %	83 %	
4	U[0.5, 7.5]	U[0.0, 5.0]	TCRAPST	201	0	8,4	60,14	3,05	7,06	2,11	4,95	4,8
			TCRAP	56 062	4	9,4	54,80	3,53	20,40	15,93	4,47	
			TCRAP/TCRAPST	27897 %	0	112 %	91 %	116 %	289 %	754 %	90 %	
5	U[6.5, 7.5]	U[0.0, 1.0]	TCRAPST	78 256	0	11,4	86,61	3,87	4,59	0,45	4,13	9,4
			TCRAP	734 265	5	12	84,47	6,60	11,53	10,13	1,40	
			TCRAP/TCRAPST	938 %	0	105 %	98 %	171 %	251 %	2241 %	34 %	
6	U[6.5, 7.5]	U[0.0, 5.0]	TCRAPST	614 247	0	12	87,89	5,41	8,11	5,52	2,59	5,2
			TCRAP	734 265	5	12	84,47	6,60	11,53	10,13	1,40	
			TCRAP/TCRAPST	120 %	0	100 %	96 %	122 %	142 %	184 %	54 %	
7	U[4.0, 5.0]	U[0.0, 1.0]	TCRAPST	18	0	8	56,88	0,91	7,12	0,03	7,09	6,6
			TCRAP	456 487	5	12	54,47	4,10	41,53	37,63	3,90	
			TCRAP/TCRAPST	2539015 %	0	150 %	96 %	450 %	583 %	125433 %	55 %	
8	U[4.0, 5.0]	U[0.0, 5.0]	TCRAPST	214	0	8,6	63,28	4,08	5,52	1,60	3,92	5
			TCRAP	456 487	5	12	54,47	4,10	41,53	37,63	3,90	
			TCRAP/TCRAPST	213345 %	0	140 %	86 %	101 %	752 %	2358 %	99 %	
9	U[3.0, 4.0]	U[0.0, 1.0]	TCRAPST	1	0	6	44,90	4,93	3,10	0,03	3,07	4,8
			TCRAP	1	5	6	44,39	6,68	3,61	2,29	1,32	
			TCRAP/TCRAPST	128 %	0	100 %	99 %	136 %	116 %	6747 %	43 %	
10	U[3.0, 4.0]	U[0.0, 5.0]	TCRAPST	6	0	7,2	50,81	4,22	6,79	3,02	3,78	2,6
			TCRAP	891	5	8,8	44,89	3,31	25,51	20,81	4,69	
			TCRAP/TCRAPST	15176 %	0	122 %	88 %	78 %	376 %	690 %	124 %	

In problem sets 3 and 4, savings in the number of activated resources occurred with a wide variation of task lengths. In the TCRAP solutions, resources working in more than one task typically had one longer and one shorter task. Still in many cases, workers in the TCRAP solutions worked in only one task, when the TCRAPST was able to allocate workers for many shared tasks.

The problem set 10 also shows savings in the number of activated resources, but the problem set 9 does not even though their tasks lengths are generated from the same distribution. In problem set 10 tasks and distances are too long for the TCRAP assign two tasks on a worker and only one 3 – 4 hours task per activated worker leaves much slack. The TCRAPS, on the other hand, routes workers through several split tasks bringing substantial savings in the number of activated resources needed. Figure 6 shows two examples of TCRAP solutions for problem set 10 and Figure 7 shows two TCRAPST solutions for the same problems. In problem set 10, traveling distances are to long for the TCRAP to assign more than one worker per task.

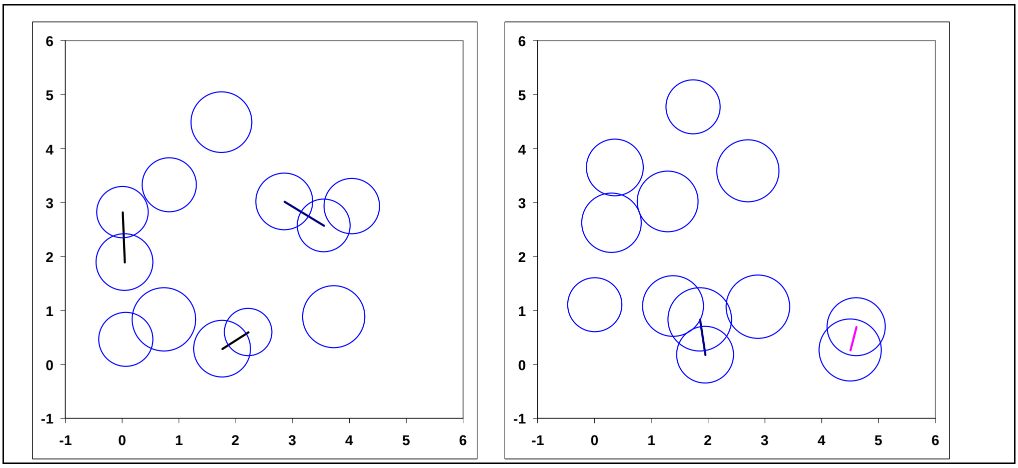

Figure 6. Two examples of the TCRAP solutions from the problem set 10

By splitting tasks between more workers, the TCRAPST can create savings in the number of active workers needed. In the two examples of TCRAPST solutions to problem set 10 in Figure 7 , all workers are assigned to work in more than one task.

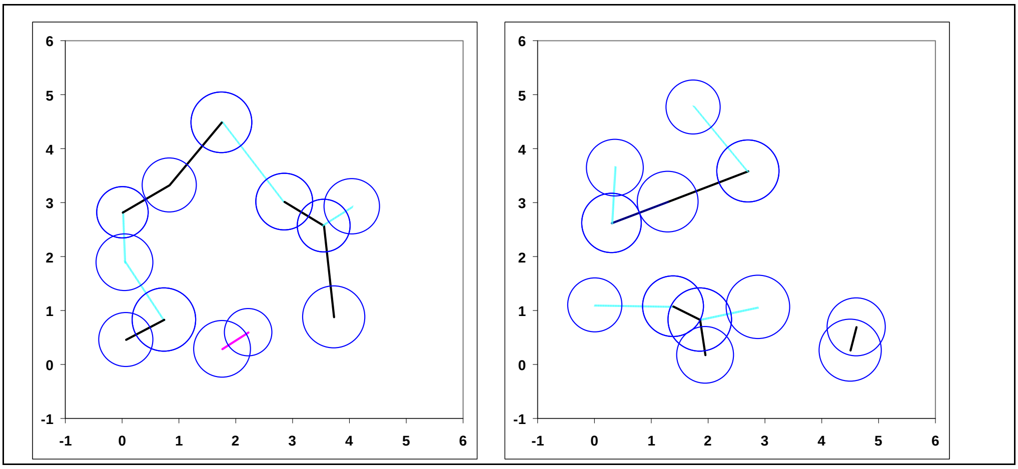

Figure 7. Two examples of the TCRAPST solutions from the problem set 10

In problem set 9, tasks are just below half of the capacity and distances between tasks are short. This allows the TCRAP to allocate practically all workers to work in two tasks because traveling between tasks does not consume much capacity. With short distances and task length distribution U[3.0, 4.0], working in two tasks leaves enough capacity for traveling between two tasks. On the other hand, there is not much savings potential in using split tasks because, on average, only a little proportion of the capacity of activated workers is left idle after they have worked in two tasks. As a result, with only 12 tasks, problem set 9 exhibited no split task generated saving in the number of activated resources. When distances become longer, as in problem set 10, one worker that could complete two tasks in TCRAP solution does not have enough total capacity for the traveling between those two tasks.

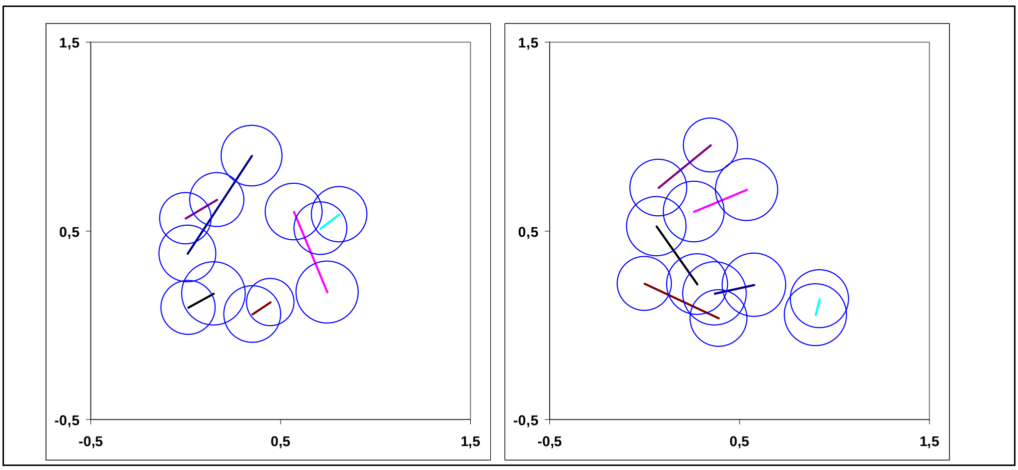

Figure 8. Two examples of the TCRAP solutions from the problem set 9

Figure 8 illustrates two TCRAP solutions in the problem set 9. Six path lines indicate, that in these cases all resources move once and only six resources are needed. Figure 6 and Figure 8 highligth the effect of increased average distance between tasks when task splitting is not allowed. In problem set 9 in Figure 8, task lengths are the same as in problem set 10 but the distance between tasks is five times longer in problem set 10 than in problem set 9.

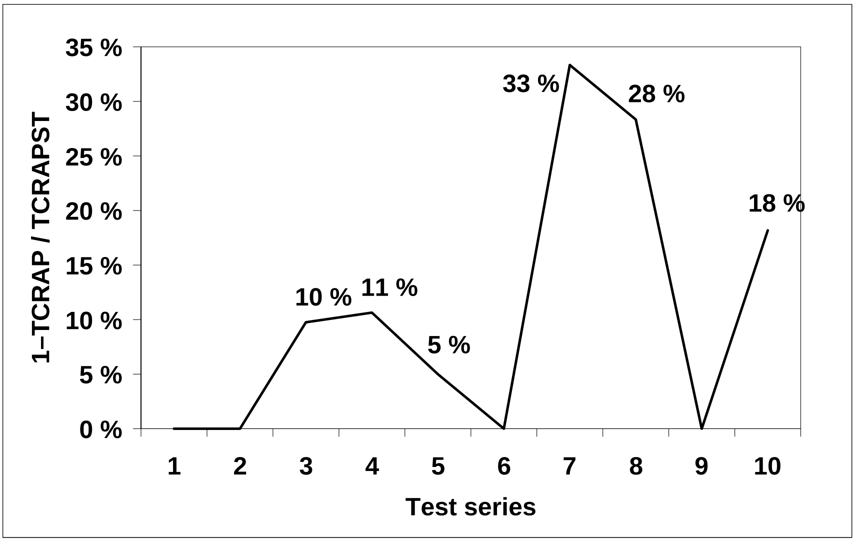

Figure 9. Savings in the average number of workers when the TCRAPST is used instead of the TCRAP

In interpreting the test problem solutions, the assumption is that every activated worker has to be paid for paid for an 8-hour workday. According to Figure 9, task splitting based capacity time savings of up to 50% are not reached in test problems, but the 33% savings in the number of activated workers in problem set 7 are still significant.

Figure 10 draws the average slack hours in problem sets. In this calculation, all idle time left from activated workers is considered as slack. For example, if 10 workers of 12 are activated, *slack* = 10 * 8 – (*work* + *traveling*). TCRAP solutions in problem sets 7 and 8 indicate a high number of slack hours because working days are 8 hours and all 12 workers are assigned to only one 4–5 hours long task per 8 hour day. So, every worker is left with 3–4 hours of slack. TCRAP solutions in the problem set 10 inlcude a lot of slack because the long distances between tasks make it difficult to allocate two 3–4 hour long tasks to every active worker.

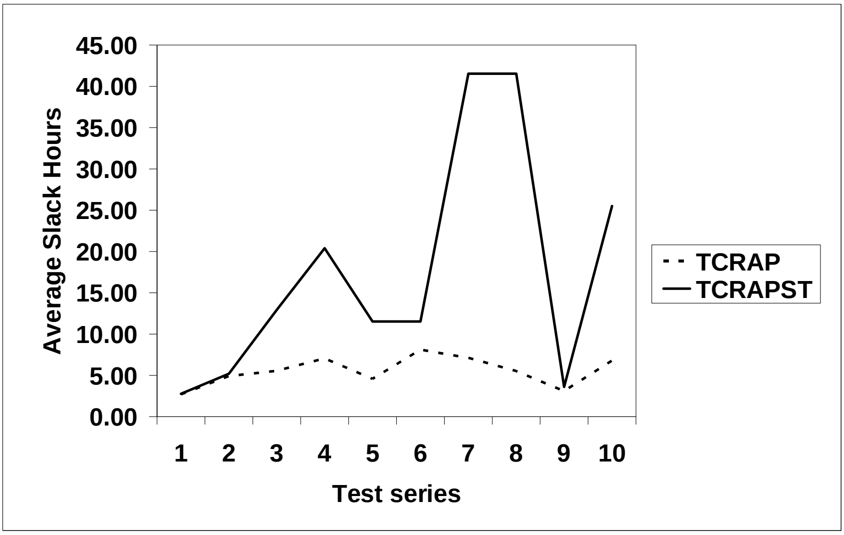

Figure 10. Average slack hours in test problems

As we compare Figure 5 and Figure 10, we can see that the biggest savings from splits correlate with the biggest TCRAP solution slacks. Figure 11 compares the number of slack hours as percentages. Using the TCRAPST seems to be efficient in removing slack in problem sets 3, 4, 5, 7, 8, and 10 where savings in slack hours is about 50%–80%. As we consider savings in slack hours, however, we should remember that the bigger absolute savings in slack hours also generate bigger monetary savings. Percentage savings in two problem sets can be the same but there can be a big difference in absolute savings. That becomes evident when we compare problem sets 5 and 6 with problem sets 7 and 8 in Figure 10 and Figure 11.

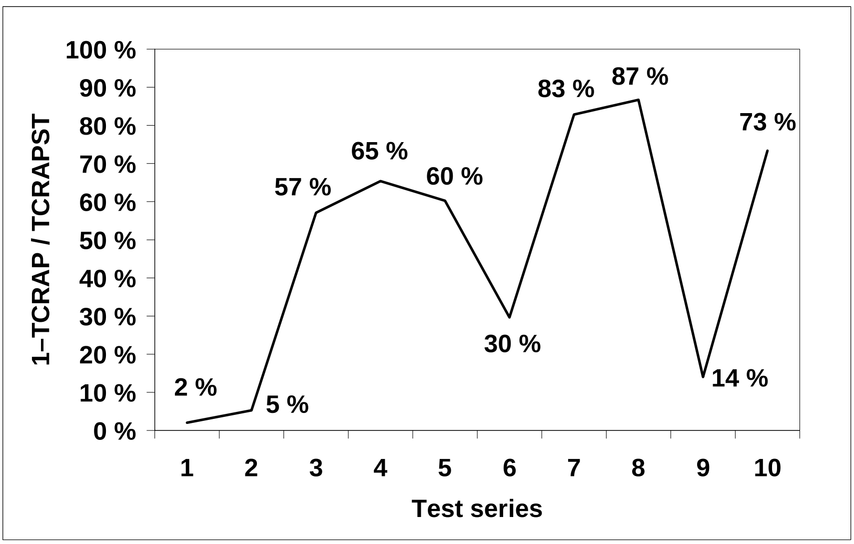

Figure 11. Savings in slack hours when the TCRAPST is used instead of the TCRAP

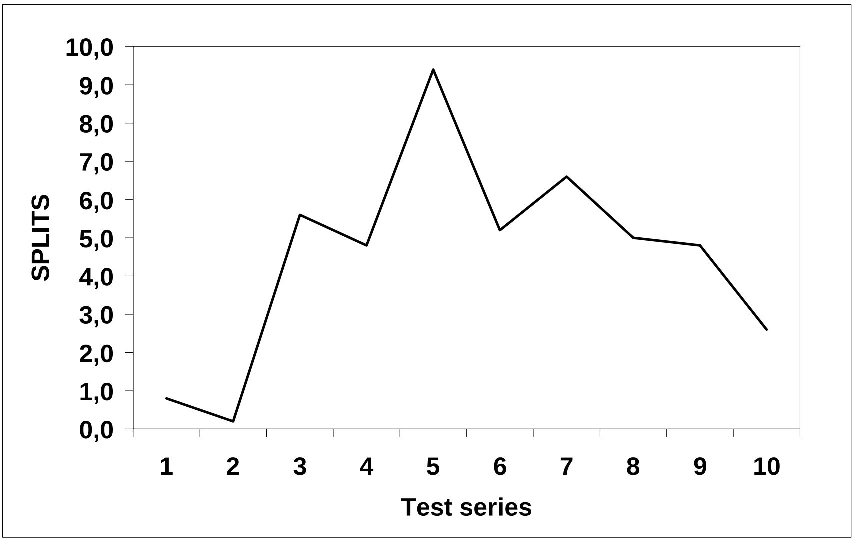

Figure 12. Average number of the TCRAPST splits in test problems

Figure 12 highlights the number of splits in the problems sets indicating that the number of splits does not always have clear correlation with the savings. For example, problem set 5 having only minor savings in the number of activated workers had the highest number of splits. Table 10 the number of workers in all 60 tasks of each problem set. The maximum number of workers per task in test problem solutions was 3, but in 4 sets of all 10 sets task splits were between two workers only. Problem sets with short distances between tasks seem to have more splits than problem sets with long distances between tasks. These observations may be of interest when developing heuristic solution methods for the TCRAPST in future.

Table 10. Number of workers in 60 tasks in each of the 10 problem sets

Number of Workers per Task	Frequency										Total Dist. U[0, 1] sets	Total Dist. U[0, 5] sets
	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9	Set 10		
1	57	59	35	37	21	38	31	35	36	47	180	216
2	2	1	22	22	31	18	25	25	24	13	104	79
3	1	0	3	1	8	4	4	0	0	0	16	5
1	95 %	98 %	58 %	62 %	35 %	63 %	52 %	58 %	60 %	78 %	60 %	72 %
2	3 %	2 %	37 %	37 %	52 %	30 %	42 %	42 %	40 %	22 %	35 %	26 %
3	2 %	0 %	5 %	2 %	13 %	7 %	7 %	0 %	0 %	0 %	5 %	2 %

5.6. Notions on Test Results

Two interesting solution features deserve more attention: First, the compression of solutions seems sometimes surprisingly weak. Second, CPLEX sometimes determines a solution as optimal even when it is not.

Weak Compression

Solutions tend to become more compresses with more solving time. Table 11 shows two TCRAPST solutions and one optimal TCRAP solution for the same problem in the problem set 9. To get the less compressed TCRAPST solution, a CPLEX option `treememlim` = 100 MB was used. For the other TCRAPST solution, `treememlim` = 500 MB was used. With a bigger `treememlim` value CPLEX searches through more solutions than with a smaller value. In both cases, the objective function value in the TCRAPST solution was lower than in the TCRAP solution. However, the less compressed TCRAPST solution uses one more resource than the TCRAP.

Running CPLEX with the option `treememlim` = 500 MB takes hours. A more practical approach may be to use the TCRAPST with a smaller `treememlim` value to get a pretty good solution and then do some manual adjustments if needed. For example, with the problem in Table 11 the small 0.07 hours of work of resource 7 can possibly be passed to resources 1,2,3 or 4 as overtime work.

Should we then add a fixed cost of activating an additional worker to improve the TCRAPST solution compression? In the test problems, different and variable working time based costs for different workers were used and there was no step-wise jump in objective function value when an additional worker was activated. Therefore, for example, adding 0.07 hours of work to be performed by the worker number 7 in Table 11 can be justified by savings in traveling costs by one of the other activated workers. Interestingly, some experiments with fixed costs included less compressed solutions than the current modeling approach without fixed costs. Other experiment with fixed costs constrained the solutions space too much by requiring so many subtour constraints that CPLEX finally failed in finding new feasible initial solutions after adding those constraints.

In split task modeling it is difficult to find a reasonable way to include fixed worker costs into the model. Instead of using fixed costs to avoid very short working days, we can add a constraint that sets a minimum duration of work for each worker in each task. That kind of minimum work time constraint requires an introduction of a dummy task with zero requirements into the TCRAPST problem formulation. Non-activated workers are then routed through the dummy task without minimum working time constraint.

Table 11. TCRAPST and TCRAP solutions with different branch-and-bound tree sizes

	Working + Traveling Hours per Resource		
Resource	**TCRAPST**		**TCRAP**
	b&b tree 100	**b&b tree 500**	**Optimal**
1	8.00	8.00	8.00
2	8.00	8.00	7.98
3	8.00	8.00	7.77
4	8.00	8.00	7.55
5	7.77	8.00	7.05
6	4.80	5.31	6.61
7	0.07		

Wrong Optimality

When resources are equal except for their cost, the cheaper resources should do more work than more expensive resources. If a resource with a smaller cost has smaller capacity utilization than a worker with a higher cost, we can easily improve the solution, even manually, by exchanging the working paths between resources with equal capacity.

In Table 12 there is one non-optimal TCRAPST solution and one TCRAP solution reported as optimal by CPLEX. Still, we can immediately see that they can not be optimal because more expensive resources are allocated more work than less expensive resources. Both solutions in Table 12 can be improved by manually exchanging the working paths of different resources so that cheaper resources always have higher utilization than more expensive resources.

The main reason for strange solution in TCRAPST is that the problem is not solved to optimality. The wrong utilizations of resources 1, 2 and 3 in TCRAP solution probably happens because CPLEX accepts a slight difference between the absolute optimum and a near optimum solution. If the near optimum solution is within that tolerance, it is accepted as an optimal solution by CPLEX. With the test problems, the smallest CPLEX 9.1 tolerance 1.0-e9 was applied. As in test problems an hour of worker 12 costs 100 000 and an hour of worker 1 costs 0.000001, the total effect of the “wrong” work loads of workers 1, 2, and 3 in TCRAP solution in Table 12 has practically no effect on the objective function value.

Table 12. Typical error in a solution: more expensive resources are allocated more work than less expensive resources.

	Working + Traveling Hours per Resource	
Resource	**TCRAPST**	**TCRAP**
1	8.00	**4.83**
2	8.00	**4.85**
3	8.00	4.86
4	**7.94**	4.83
5	**8.00**	4.72
6	**7.93**	4.69
7	**8.00**	4.66
8	7.86	4.53
9	2.39	4.49
10		4.43
11		4.30
12		4.28

6. Conclusions

Two models, the TCRAPST and the TCRAP, were used to solve 10 sets of problems that simulate a service environment where task requirements are measured as capacity time needed to perform a task. Each resource had the same amount of capacity that was measured as time. Capacity is used to perform tasks, to move between tasks and to stay idle. The number of resources needed to perform tasks was minimized by simultaneously maximizing the utilization of least expensive resources and minimizing the utilization of most expensive resources. The TCRAPST and the TCRAP were used to generate plans that route and allocate resources for each task. When the TCRAP required each task to be completed by one resource, the TCRAPST allowed more resources to work on the same task by splitting the work loads between resources.

The study showed that the TCRAPST can generate more efficient plans than the TCRAP. Most savings appear when the average length of tasks is just over half of resource capacity and the average distance between tasks is short. In such conditions, the TCRAP can allocate only one resource per task and almost half of the capacity of that resource stays unused. The TCRAPST, on the other hand, can generate solutions where most resources are either fully used or completely idle.

Tests with synthetic problems showed the difficulty of finding optimal solutions to realistic sized problems. As further research, more efficient solution methods should be developed and realistic applications should be found.

List of References

Archetti, C. & Speranza, M.G. (2008) "The Split Delivery Vehicle Routing Problem: A Survey", In: Golden, B., Raghavan, S. & Wasil, E.(eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges,* Operations Research/Computer Science Interfaces Series, Vol. 43, pp. 103-122, Springer, US.

Dror, M. & Trudeau, P. (1989) "Savings by Split Delivery Routing", *Transportation Science,* Vol. 23, No. 2, pp. 141-145.

Dror, M. & Trudeau, P. (1987) *Split Delivery Routing,* Publication No. 507 (January), Centre de recherche sur les transports, Université de Montréal, Canada.

Porkka, P.P. (2009) *Modeling Time Capacitated Resource Allocation In Services Allowing For Split Tasks — Vehicle Routing Problem Approach,* Unpublished Working Paper in the Doctoral Dissertation of Pasi P. Porkka, Helsinki School of Economics, Helsinki, Finland.

Toth, P. & Vigo, D. (2001) "The vehicle routing problem", *SIAM monographs on discrete mathematics and applications,* Philadelphia, US: Society for Industrial and Applied Mathematics, USA.

APPENDIX: The TCRAP as an AMPL Model

```
# SETS AND PARAMETERS
# ------------------------------------------------------------------------------------
set TASK ordered;                                         # ordered set of tasks
set WORKER      ;                                         # set of resources
set TRANSF = {i in TASK, j in TASK: ord(i) < ord(j)};     # allowed transfers between
                                                          # customer facilities

param capacity {WORKER} = 8;              # capacity of resource (here 8 assumed)
param distance {TASK, TASK} >= 0;         # set-up time between tasks
param requirement {TASK};                 # capacity required by task
param workercost {WORKER};                # cost of using resource


# VARIABLES
# ------------------------------------------------------------------------------------
var Place {WORKER, TASK} binary;          # 1, if resource visits task; 0 otherwise
var Route {WORKER, TRANSF} binary;        #  1,  if  resource  moves  between  two  tasks;  1
otherwise
var Beginning {WORKER, TASK} binary;      # determines the first task of a resource
var End {WORKER, TASK} binary;            # determines the last task of a worker
var WorkTime {WORKER, TASK};              # capacity time of resource used in task


# OBJECTIVE FUNCTION
# ------------------------------------------------------------------------------------
minimize Worker_Time_Needed:
        sum{w in WORKER}(
                sum{(i,j) in TRANSF}Route[w,i,j]*distance[i,j]
                        + sum{f in TASK}Place[w,f]*requirement[f])*workercost[w];


# CAPACITY CONSTRAINTS
# -------------------------------------------------------------------------------------
# ALL WORK HAS TO BE DONE
subject to WorkDemand {i in TASK: i<>"DRAIN"}:
        sum {w in WORKER} Place[w,i] = 1;

# WORK, TRAVELING AND SLACK CAN NOT EXCEED CAPACITY
subject to WorkSupply {w in WORKER}:
        sum {i in TASK} WorkTime[w,i]
        + sum {(i,j) in TRANSF} Route[w,i,j] * distance[i,j] <= capacity[w];

# FOR WORK, A WORK ASSIGNMENT DECISION HAS TO BE MADE
subject to Worker_Time {w in WORKER, i in TASK} :
        WorkTime[w,i] = Place[w,i]*requirement[i] ;
```

```
# INTRAPERIOD SET-UP/TRAVELING CONSTRAINTS
# ------------------------------------------------------------------------------------
# 1) MOVEMENTS CAN ONLY TAKE PLACE IF ALLOCATION DECISION HAS BEEN MADE
# 2) A RESOURCE HAS TO BE ALLOCATED TO AT LEAST ONE TASK
subject to Work_For_Traveling {w in WORKER, f in TASK}:
        Place[w,f]
        -   (sum{j   in   TASK:   ord(j)<ord(f)}   Route[w,j,f]   +   sum{j   in   TASK:
ord(j)>ord(f)}Route[w,f,j]
        + Beginning[w,f] + End[w,f])/2 = 0 ;


# IN AND OUT OF THE SYSTEM
# ------------------------------------------------------------------------------------

subject to Worker_Has_To_Come_From_Somewhere {w in WORKER} :
        sum{f in TASK} Beginning[w,f] = 1 ;

subject to Worker_Has_To_Go_Somewhere {w in WORKER} :
        sum{f in TASK} End[w,f] = 1 ;


# SUBTOUR CONSTRAINTS
# ------------------------------------------------------------------------------------
```

HELSINGIN KAUPPAKORKEAKOULUN JA AALTO-YLIOPISTON KAUPPAKORKEAKOULUN JULKAISUJA
Publications of the Helsinki School of Economics

A-SARJA: VÄITÖSKIRJOJA - DOCTORAL DISSERTATIONS. ISSN 1237-556X.

A:310. JARI PAULAMÄKI: Kauppiasyrittäjän toimintavapaus ketjuyrityksessä. Haastattelututkimus K-kauppiaan kokemasta toimintavapaudesta agenttiteorian näkökulmasta. 2008. Korjattu painos. ISBN 978-952-488-246-0, E-version: 978-952-488-247-7.

A:321. BRETT FIFIELD: A Project Network: An Approach to Creating Emergent Business. 2008. ISBN 978-952-488-206-4, E-version: 978-952-488-207-1.

A:322. ANTTI NURMI: Essays on Management of Complex Information Systems Development Projects. 2008. ISBN 978-952-488-226-2.

A:323. SAMI RELANDER: Towards Approximate Reasoning on New Software Product Company Success Potential Estimation. A Design Science Based Fuzzy Logic Expert System. 2008. ISBN 978-952-488-227-9.

A:324. SEPPO KINKKI: Essays on Minority Protection and Dividend Policy. 2008. ISBN 978-952-488-229-3.

A:325. TEEMU MOILANEN: Network Brand Management: Study of Competencies of Place Branding Ski Destinations. 2008. ISBN 978-952-488-236-1.

A:326. JYRKI ALI-YRKKÖ: Essays on the Impacts of Technology Development and R&D Subsidies. 2008. ISBN 978-952-488-237-8.

A:327. MARKUS M. MÄKELÄ: Essays on software product development. A Strategic management viewpoint. 2008. ISBN 978-952-488-238-5.

A:328. SAMI NAPARI: Essays on the gender wage gap in Finland. 2008. ISBN 978-952-488-243-9.

A:329. PAULA KIVIMAA: The innovation effects of environmental policies. Linking policies, companies and innovations in the Nordic pulp and paper industry. 2008. ISBN 978-952-488-244-6.

A:330. HELI VIRTA: Essays on Institutions and the Other Deep Determinants of Economic Development. 2008. ISBN 978-952-488-267-5.

A:331. JUKKA RUOTINEN: Essays in trade in services difficulties and possibilities. 2008. ISBN 978-952-488-271-2, E-version: ISBN 978-952-488-272-9.

A:332. IIKKA KORHONEN: Essays on commitment and government debt structure. 2008. ISBN 978-952-488-273-6, E-version: ISBN 978-952-488-274-3.

A:333. MARKO MERISAVO: The interaction between digital marketing communication and customer loyalty. 2008. ISBN 978-952-488-277-4, E-version 978-952-488-278-1.

A:334. PETRI ESKELINEN: Reference point based decision support tools for interactive multiobjective optimization. 2008. ISBN 978-952-488-282-8.

A:335. SARI YLI-KAUHALUOMA: Working on technology: a study on collaborative R&D work in industrial chemistry. 2008. ISBN 978-952-488-284-2

A:336. JANI KILPI: Sourcing of availability services - case aircraft component support. 2008. ISBN 978-952-488-284-2, 978-952-488-286-6 (e-version).

A:337. HEIDI SILVENNOINEN: Essays on household time allocation decisions in a collective household model. 2008. ISBN 978-952-488-290-3, ISBN 978-952-488-291-0 (e-version).

A:338. JUKKA PARTANEN: Pk-yrityksen verkostokyvykkyydet ja nopea kasvu - case: Tiede- ja teknologiavetoiset yritykset. 2008. ISBN 978-952-488-295-8.

A:339. PETRUS KAUTTO: Who holds the reins in Integrated Product Policy? An individual company as a target of regulation and as a policy maker. 2008. ISBN 978-952-488-300-9, 978-952-488-301-6 (e-version).

A:340. KATJA AHONIEMI: Modeling and Forecasting Implied Volatility. 2009. ISBN 978-952-488-303-0, E-version: 978-952-488-304-7.

A:341. MATTI SARVIMÄKI: Essays on Migration. 2009. ISBN 978-952-488-305-4, 978-952-488-306-1 (e-version).

A:342. LEENA KERKELÄ: Essays on Globalization – Policies in Trade, Development, Resources and Climate Change. 2009. ISBN 978-952-488-307-8, E-version: 978-952-488-308-5.

A:343. ANNELI NORDBERG: Pienyrityksen dynaaminen kyvykkyys - Empiirinen tutkimus graafisen alan pienpainoyrityksistä. 2009. ISBN 978-952-488-318-4.

A:344. KATRI KARJALAINEN: Challenges of Purchasing Centralization – Empirical Evidence from Public Procurement. 2009. ISBN 978-952-488-322-1, E-version: 978-952-488-323-8.

A:345. JOUNI H. LEINONEN: Organizational Learning in High-Velocity Markets. Case Study in The Mobile Communications Industry. 2009. ISBN 978-952-488-325-2.

A:346. JOHANNA VESTERINEN: Equity Markets and Firm Innovation in Interaction. - A Study of a Telecommunications Firm in Radical Industry Transformation. 2009. ISBN 978-952-488-327-6, E-version: 978-952-488-374-0.

A:347. JARI HUIKKU: Post-Completion Auditing of Capital Investments and Organizational Learning. 2009. ISBN 978-952-488-334-4, E-version: 978-952-488-335-1.

A:348. TANJA KIRJAVAINEN: Essays on the Efficiency of Schools and Student Achievement. 2009. ISBN 978-952-488-336-8, E-version: 978-952-488-337-5.

A:349. ANTTI PIRJETÄ: Evaluation of Executive Stock Options in Continuous and Discrete Time. 2009. ISBN 978-952-488-338-2, E-version: 978-952-488-339-9.

A:350. OLLI KAUPPI: A Model of Imperfect Dynamic Competition in the Nordic Power Market. 2009. ISBN 978-952-488-340-5, E-version: 978-952-488-341-2.

A:351. TUIJA NIKKO: Dialogic Construction of Understanding in Cross-border Corporate Meetings. 2009. ISBN 978-952-488-342-9, E-version: 978-952-488-343-6.

A:352. MIKKO KORIA: Investigating Innovation in Projects: Issues for International Development Cooperation. 2009. ISBN 978-952-488-344-3, E-version: 978-952-488-345-0.

A:353. MINNA MUSTONEN: Strategiaviestinnän vastaanottokäytännöt - Henkilöstö strategia-viestinnän yleisönä. 2009. ISBN 978-952-488-348-1, E-versio: 978-952-488-349-8.

A:354. MIRELLA LÄHTEENMÄKI: Henkilötietojen hyödyntäminen markkinoinnissa kuluttajien tulkitsemana. Diskurssianalyyttinen tutkimus kuluttajan tietosuojasta. 2009. ISBN 978-952-488-351-1, E-versio: 978-952-488-352-8.

A:355. ARNO KOURULA: Company Engagement with Nongovernmental Organizations from a Corporate Responsibility Perspective. 2009. ISBN 978-952-488-353-5, E-version: 978-952-488-354-2.

A:356. MIKA WESTERLUND: Managing Networked Business Models: Essays in the Software Industry. 2009. ISBN 978-952-488-363-4

A:357. RISTO RAJALA: Determinants of Business Model Performance in Software Firms. 2009. ISBN 978-952-488-369-6, E-version: 978-952-488-370-2.

A:358. PASI P. PORKKA: Capacitated Timing of Mobile and Flexible Service Resources. 2010 ISBN 978-952-60-1000-7, E-version: ISBN 978-952-60-1001-4.

B-SARJA: TUTKIMUKSIA - RESEARCH REPORTS. ISSN 0356-889X.

B:84. PÄIVI KARHUNEN – RIITTA KOSONEN – JOHANNA LOGRÉN – KRISTO OVASKA: Suomalaisyritysten strategiat Venäjän muuttuvassa liiketoimintaympäristössä. 2008. ISBN 978-953-488-212-5, E-versio: 978-952-488-241-5.

B:85. MARJA MATTILA – EEVA KEROLA – RIITTA KOSONEN: Unkari suomalaisyritysten toimintaympäristönä. 2008. ISBN 978-952-488-213-2, E-versio: 978-952-488-222-4.

B:86. KRISTIINA KORHONEN – ANU PENTTILÄ – MAYUMI SHIMIZU – EEVA KEROLA – RIITTA KOSONEN: Intia suomalaisyritysten toimintaympäristönä.2008. ISBN 978-952-488-214-9, E-versio: 978-952-488-283-5

B:87. SINIKKA VANHALA – SINIKKA PESONEN: Työstä nauttien. SEFE:en kuuluvien nais- ja miesjohtajien näkemyksiä työstään ja urastaan. 2008. ISBN 978-952-488-224-8, E-versio: 978-952-488-225-5.

B:88. POLINA HEININEN – OLGA MASHKINA – PÄIVI KARHUNEN – RIITTA KOSONEN: Leningradin lääni yritysten toimintaympäristönä: pk-sektorin näkökulma. 2008. ISBN 978-952-488-231-6, E-versio: 978-952-488-235-4.

B:89. Ольга Машкина – Полина Хейнинен: Влияние государственного сектора на развитие малого и среднего предпринимательства в Ленинградской области: взгляд предприятий.2008. ISBN 978-952-488-233-0, E-version: 978-952-488-240-8.

B:90. MAI ANTTILA – ARTO RAJALA (Editors): Fishing with business nets – keeping thoughts on the horizon Professor Kristian Möller. 2008. ISBN 978-952-488-249-1, E-version: 978-952-488-250-7.

B:91. RENÉ DE KOSTER – WERNER DELFMANN (Editors): Recent developments in supply chain management. 2008. ISBN 978-952-488-251-4, E-version: 978-952-488-252-1.

B:92. KATARIINA RASILAINEN: Valta orkesterissa. Narratiivinen tutkimus soittajien kokemuksista ja näkemyksistä. 2008.
ISBN 978-952-488-254-5, E-versio: 978-952-488-256-9.

B:93. SUSANNA KANTELINEN: Opiskelen, siis koen. Kohti kokevan subjektin tunnistavaa korkeakoulututkimusta. 2008. ISBN 978-952-488-257-6, E-versio: 978-952-488-258.

B:94. KATRI KARJALAINEN – TUOMO KIVIOJA – SANNA PELLAVA: Yhteishankintojen kustannusvaikutus. Valtion hankintatoimen kustannussäästöjen selvittäminen. 2008.
ISBN 978-952-488-263-7, E-versio: ISBN 978-952-488-264-4.

B:95. ESKO PENTTINEN: Electronic Invoicing Initiatives in Finland and in the European Union – Taking the Steps towards the Real-Time Economy. 2008.
ISBN 978-952-488-268-2, E-versio: ISBN 978-952-488-270-5.

B:96. LIISA UUSITALO (Editor): Museum and visual art markets. 2008.
ISBN 978-952-488-287-3, E-version: ISBN 978-952-488-288-0.

B:97. EEVA-LIISA LEHTONEN: Pohjoismaiden ensimmäinen kauppatieteiden tohtori Vilho Paavo Nurmilahti 1899-1943. 2008. ISBN 978-952-488-292-7,
E-versio: ISBN 978-952-488-293-4.

B:98. ERJA KETTUNEN – JYRI LINTUNEN – WEI LU – RIITTA KOSONEN: Suomalaisyritysten strategiat Kiinan muuttuvassa toimintaympäristössä. 2008 ISBN 978-952-488-234-7,
E-versio: ISBN 978-952-488-297-2.

B:99. SUSANNA VIRKKULA – EEVA-KATRI AHOLA – JOHANNA MOISANDER – JAAKKO ASPARA – HENRIKKI TIKKANEN: Messut kuluttajia osallistavan markkinakulttuurin fasilitaattorina: messukokemuksen rakentuminen Venemessuilla. 2008.
ISBN 978-952-488-298-9, E-versio: ISBN 978-952-488-299-6.

B:100. PEER HULL KRISTENSEN – KARI LILJA (Eds): New Modes of Globalization: Experimentalist Forms of Economics Organization and Enabling Welfare Institutions – Lessons from The Nordic Countries and Slovenia. 2009. ISBN 978-952-488-309-2,
E-version: 978-952-488-310-8.

B:101. VIRPI SERITA – ERIK PÖNTISKOSKI (eds.)
SEPPO MALLENIUS – VESA LEIKOS – KATARIINA VILLBERG – TUUA RINNE – NINA YPPÄRILÄ – SUSANNA HURME: Marketing Finnish Design in Japan. 2009.
ISBN 978-952-488-320-7. E-version: ISBN 978-952-488-321-4.

B:102. POLINA HEININEN – OLLI-MATTI MIKKOLA – PÄIVI KARHUNEN – RIITTA KOSONEN: Yritysrahoitusmarkkinoiden kehitys Venäjällä. Pk-yritysten tilanne Pietarissa. 2009.
ISBN 978-952-488-329-0. E-version: ISBN 978-952-488-331-3.

B:103. ARTO LAHTI: Liiketoimintaosaamisen ja yrittäjyyden pioneeri Suomessa. 2009.
ISBN 978-952-488-330-6.

B:104. KEIJO RÄSÄNEN: Tutkija kirjoittaa - esseitä kirjoittamisesta ja kirjoittajista akateemisessa työssä. 2009. ISBN 978-952-488-332-0. E-versio: ISBN 978-952-488-333-7.

B:105. TIMO EKLUND – PETRI JÄRVIKUONA – TUOMAS MÄKELÄ – PÄIVI KARHUNEN: Kazakstan suomalaisyritysten toimintaympäristönä. 2009. ISBN 978-952-488-355-9.

B:106. ARTO LINDBLOM – RAMI OLKKONEN – VILJA MÄKELÄ (TOIM.): Liiketoimintamallit, innovaatiotoiminta ja yritysten yhteistyön luonne kaupan arvoketjussa.2009. ISBN 978-952-488-356-6. E-versio: ISBN 978-952-488-357-3.

B:107. MIKA GABRIELSSON – ANNA SALONEN – PAULA KILPINEN – MARKUS PAUKKU – TERHI VAPOLA – JODY WREN – LAURA ILONEN – KATRIINA JUNTUNEN: Responding to Globalization: Strategies and Management for Competitiveness. Final Report of a TEKES-project 1.8.2006-30.4.2009. 2009. ISBN 978-952-488-362-7.

B:108. MATTI ROSSI – JONATHAN SPRINKLE – JEFF GRAY – JUHA-PEKKA TOLVANEN (EDS.) Proceedings of the 9th OOPSLA Workshop on Domain-Specific Modeling (DSM'09). 2009. ISBN 978-952-488-371-9. E--version: ISBN 978-952-488-372-6.

B:109. LEENA LOUHIALA-SALMINEN – ANNE KANKAANRANTA (Editors): The Ascent of International Business Communication. 2009. ISBN 978-952-488-373-3. E-version: ISBN 978-952-488-381-8.

B:110. MIKKO VIEMERÖ: Tietosuoja sähköisessä kaupassa ja sähköisessä viestinnässä. 2009. ISBN 978-952-488-376-4.

B:111. RAIJA JÄRVINEN – JUHA UUSPELTO: Uhkaavatko asiakkaat? Kaupan henkilökunnan näkemyksiä turvallisuusuhkista. 2009. ISBN 978-952-488-379-5. E-versio: ISBN 978-952-488-380-1.

B:113. LIISA UUSITALO – MIRELLA LÄHTEENMÄKI (TOIM.): Kuluttaja ja media tietotaloudessa. 2009. ISBN 978-952-488-383-2. E-versio: ISBN 978-952-488-384-9

B:114. REIJO LUOSTARINEN: kansainvälinen liiketoiminta ja sen rooli Helsingin kauppakorkeakoulun kansainvälistymisessä. 2009. ISBN 978-952-60-1002-1.

N-SARJA: HELSINKI SCHOOL OF ECONOMICS. MIKKELI BUSINESS CAMPUS PUBLICATIONS. ISSN 1458-5383

N:74. MIKKO SAARIKIVI: Pk-yritysten kansainvälistymisen sopimukset. 2008. ISBN 978-952-488-210-1.

N:75. LAURA TUUTTI: Uutta naisjohtajuutta Delfoi Akatemiasta – hankkeen vaikuttavuus. 2008. ISBN 978-952-488-211-8.

N:76. LAURA KEHUSMAA – JUSSI KÄMÄ – ANNE GUSTAFSSON-PESONEN (ohjaaja): StuNet -Business Possibilities and Education - hankkeen arviointi. 2008. ISBN 978-952-488-215-6.

N:77. PÄIVI KARHUNEN – ERJA KETTUNEN – VISA MIETTINEN – TIINAMARI SIVONEN: Determinants of knowledge-intensive entrepreneurship in Southeast Finland and Northwest Russia. 2008. ISBN 978-952-488-223-1.

N:78. ALEKSANDER PANFILO – PÄIVI KARHUNEN – VISA MIETTINEN: Suomalais-venäläisen innovaatioyhteistyön haasteet toimijanäkökulmasta. 2008. ISBN 978-952-488-232-3.

N:79. VESA KOKKONEN: Kasva Yrittäjäksi – koulutusohjelman vaikuttavuus. 2008. ISBN 978-952-488-248-4.

N:80. VESA KOKKONEN: Johtamisen taidot - hankkeessa järjestettyjen koulutusohjelmien vaikuttavuus. 2008. ISBN 978-952-488-259-0.

N:81. MIKKO SAARIKIVI: Raportti suomalaisten ja brittiläisten pk-yritysten yhteistyön kehittämisestä uusiutuvan energian sektorilla. 2008. ISBN 978-952-488-260-6.

N:82. MIKKO SAARIKIVI – JARI HANDELBERG – TIMO HOLMBERG – ARI MATILAINEN: Selvitys lujitemuovikomposiittituotteiden mahdollisuuksista rakennusteollisuudessa. 2008. ISBN 978-952-488-262-0.

N:83. PÄIVI KARHUNEN – SVETLANA LEDYAEVA – ANNE GUSTAFSSON-PESONEN – ELENA MOCHNIKOVA – DMITRY VASILENKO: Russian students' perceptions of entrepreneurship. Results of a survey in three St. Petersburg universities. Entrepreneurship development –project 2. 2008. ISBN 978-952-488-280-4.

N:84. PIIA NIKULA – ANU PENTTILÄ – OTTO KUPI – JUHANA URMAS – KIRSI KOMMONEN: Sirpaleisuudesta kilpailukyvyn keskiöön Asiantuntijoiden näkemyksiä luovien alojen kansainvälistymisestä. 2009. ISBN 978-952-488-346-7.

N:85 JUHANA URMAS – OTTO KUPI – PIIA NIKULA – ANU PENTTILÄ – KIRSI KOMMONEN: " Kannattaa ottaa pienikin siivu" – Luovien alojen yritysten näkemyksiä kansainvälistymisestä. 2009. ISBN 978-952-488-347-4.

W-SARJA: TYÖPAPEREITA - WORKING PAPERS . ISSN 1235-5674.
ELECTRONIC WORKING PAPERS, ISSN 1795-1828.

W:440. KALYANMOY DEB – KAISA MIETTINEN – SHAMIK CHAUDHURI: Estimating Nadir Objective Vector: Hybrid of Evolutionary and Local Search. 2008. ISBN 978-952-488-209-5.

W:441. ARTO LAHTI: Globalisaatio haastaa pohjoismaisen palkkatalousmallin. Onko löydettävissä uusia aktiivisia toimintamalleja, joissa Suomi olisi edelleen globalisaation voittaja? 2008. ISBN 978-952-488-216-3.

W:442. ARTO LAHTI: Semanttinen Web – tulevaisuuden internet. Yrittäjien uudet liiketoimintamahdollisuudet. 2008. ISBN 978-952-488-217-0.

W:443. ARTO LAHTI: Ohjelmistoteollisuuden globaali kasvustrategia ja immateriaalioikeudet. 2008. ISBN 978-952-488-218-7.

W:444. ARTO LAHTI: Yrittäjän oikeusvarmuus globaalisaation ja byrokratisoitumisen pyörteissä. Onko löydettävissä uusia ja aktiivisia toimintamalleja yrittäjien syrjäytymisen estämiseksi? 2008. ISBN 978-952-488-219-4.

W:445. PETRI ESKELINEN: Objective trade-off rate information in interactive multiobjective optimization methods – A survey of theory and applications. 2008. ISBN 978-952-488-220-0.

W:446. DEREK C. JONES – PANU KALMI: Trust, inequality and the size of co-operative sector – Cross-country evidence. 2008. ISBN 978-951-488-221-7.

W:447. KRISTIINA KORHONEN – RIITTA KOSONEN – TIINAMARI SIVONEN – PASI SAUKKONEN: Pohjoiskarjalaisten pienten ja keskisuurten yritysten Venäjäyhteistyöpotentiaali ja tukitarpeet. 2008. ISBN 978-952-488-228-6.

W:448. TIMO JÄRVENSIVU – KRISTIAN MÖLLER: Metatheory of Network Management: A Contingency Perspective. 2008. ISBN 978-952-488-231-6.

W:449. PEKKA KORHONEN: Setting “condition of order preservation” requirements for the priority vector estimate in AHP is not justified. 2008. ISBN 978-952-488-242-2.

W:450. LASSE NIEMI – HANNU OJALA – TOMI SEPPÄLÄ: Misvaluation of takeover targets and auditor quality. 2008. ISBN 978-952-488-255-2.

W:451. JAN-ERIK ANTIPIN – JANI LUOTO: Forecasting performance of the small-scale hybrid New Keynesian model. 2008. ISBN 978-952-488-261-3.

W:452. MARKO MERISAVO: The Interaction between Digital Marketing Communication and Customer Loyalty. 2008. ISBN 978-952-488-266-8.

W:453. PETRI ESKELINEN – KAISA MIETTINEN: Trade-off Analysis Tool with Applicability Study for Interactive Nonlinear Multiobjective Optimization. 2008. ISBN 978-952-488-269-9.

W:454. SEPPO IKÄHEIMO – VESA PUTTONEN – TUOMAS RATILAINEN: Antitakeover provisions and performance – Evidence from the Nordic countries. 2008. ISBN 978-952-488-275-0.

W:455. JAN-ERIK ANTIPIN: Dynamics of inflation responses to monetary policy in the EMU area. 2008. ISBN 978-952-488-276-7.

W:456. KIRSI KOMMONEN: Narratives on Chinese colour culture in business contexts. The Yin Yang Wu Xing of Chinese values. 2008. ISBN 978-952-488-279-8.

W:457. MARKKU ANTTONEN – MIKA KUISMA – MINNA HALME – PETRUS KAUTTO: Materiaalitehokkuuden palveluista ympäristömyötäistä liiketoimintaa (MASCO2). 2008. ISBN 978-952-488-279-8.

W:458. PANU KALMI – DEREK C. JONES – ANTTI KAUHANEN: Econometric case studies: overview and evidence from recent finnish studies. 2008. ISBN 978-952-488-289-7.

W:459. PETRI JYLHÄ – MATTI SUOMINEN – JUSSI-PEKKA LYYTINEN: Arbitrage Capital and Currency Carry Trade Returns. 2008. ISBN 978-952-488-294-1.

W:460. OLLI-MATTI MIKKOLA – KATIA BLOIGU – PÄIVI KARHUNEN: Venäjä-osaamisen luonne ja merkitys kansainvälisissä suomalaisyrityksissä. 2009. ISBN 978-952-488-302-3.

W:461. ANTTI KAUHANEN – SATU ROPONEN: Productivity Dispersion: A Case in the Finnish Retail Trade. 2009. ISBN 978-952-488-311-5.

W:462. JARI HUIKKU: Design of a Post-Completion Auditing System for Organizational Learning. 2009. ISBN 978-952-488-312-2.

W:463. PYRY-ANTTI SIITARI: Identifying Efficient Units in Large-Scale Dea Models Using Efficient Frontier Approximation. 2009. ISBN 978-952-488-313-9.

W:464. MARKKU KALLIO – MERJA HALME: Conditions for Loss Averse and Gain Seeking Consumer Price Behavior. 2009. ISBN 978-952-488-314-6.

W:465. MERJA HALME – OUTI SOMERVUORI: Study of Internet Material Use in Education in Finland. 2009. ISBN 978-952-488-315-3.

W:466. RAIMO LOVIO: Näkökulmia innovaatiotoiminnan ja –politiikan muutoksiin 2000-luvulla. 2009. ISBN 978-952-488-316-0.

W:467. MERJA HALME – OUTI SOMERVUORI: Revisiting Demand Reactions to Price Changes. 2009. ISBN 978-952-488-317-7.

W:468. SAMULI SKURNIK: SSJS Strategiabarometri – kehitystyö ja nykyvaihe. 2009. ISBN 978-952-488-319-1.

W:469. TOM RAILIO: A Brief Description of The Transdisciplinary Jurionomics and The Scandinavian Institutional Sources of Law Framework. 2009. ISBN 978-952-488-324-5.

W:470. KALYANMOY DEB – KAISA MIETTINEN – SHAMIK CHAUDHURI: An Estimation of Nadir Objective Vector Using a Hybrid Evolutionary-Cum-Local-Search Procedure. 2009. ISBN 978-952-488-326-9.

W:471. JENNI AHONEN – MARI ANTTONEN – ANTTI HEIKKINEN – JANI HÄTÄLÄ – JASMI LEHTOLA – LAURI NURMILAUKAS – TEEMU PELTOKALLIO – ANNINA PIEKKARI – MARJO REEN – SEBASTIAN SMART: Doing Business in Hungary. 2009. ISBN 978-952-488-350-4.

W:472. MIKA WESTERLUND: The role of Network Governance in Business Model Performance. 2009. ISBN 978-952-488-361-0.

W:473. DMITRY FILATOV – SINIKKA PARVIAINEN – PÄIVI KARHUNEN: The St. Petersburg Insurance Market: Current Challenges and Future Opportunities. 2009. ISBN 978-952-488-365-8.

W:474. MARKKU KALLIO – MERJA HALME: Redefining Loss Averse and Gain Seeking Consumer Price Behavior Based on Demand Response. 2009. ISBN 978-952-488-366-5.

W:475. JOHANNA BRAGGE – TUURE TUUNANEN – PENTTI MARTTIIN: Inviting Lead Users from Virtual Communities to Co-create Innovative IS Services in a Structured Groupware Environment. 2009. ISBN 978-952-488-367-2.

W:476. RISTO RAJALA: Antecedents to and Performance Effects of Software Firms' Business Models. 2009. ISBN 978-952-488-368-9.

W:477. HANNU SAARINEN: Käyttäjäinnovaatioiden mahdollisuudet ja ongelmat – tapaustutkimus peliteollisuuden käytännöistä. 2009. ISBN 978-952-488-382-5.

Kaikkia Helsingin kauppakorkeakoulun ja Aalto-yliopiston kauppakorkeakoulun julkaisusarjassa ilmestyneitä julkaisuja voi tilata osoitteella:

KY-Palvelu Oy
Kirjakauppa
Runeberginkatu 14-16
00100 Helsinki
Puh. (09) 4703 8310, fax (09) 495 617
Sähköposti: kykirja@ky.hse.fi

Aalto-yliopiston kauppakorkeakoulu
Julkaisutoimittaja
PL 1210
00101 Helsinki
Puh. (09) 4703 8579, fax (09) 4703 8305
Sähköposti: julkaisu@hse.fi

All the publications can be ordered from

Aalto University School of Economics
Publications officer
P.O.Box 1210
FIN-00101 Helsinki
Phone +358-9-4703 8579, fax +358-9-4703 8305
E-mail: julkaisu@hse.fi