Optimization Strategies for Integrated Knapsack and Traveling Salesman Problems

Andreas Beham[1,2], Judith Fechter[1,2], Michael Kommenda[1,2], Stefan Wagner[1], Stephan M. Winkler[1], and Michael Affenzeller[1,2]

[1]Heuristic and Evolutionary Algorithms Laboratory, School of Informatics, Communications and Media, University of Applied Sciences Upper Austria, Hagenberg Campus, Softwarepark 11, 4232 Hagenberg, Austria,
[2]Institute for Formal Models and Verification, Johannes Kepler University Linz, Altenberger Straße 69, 4040 Linz, Austria

July 28, 2017

Technical Report HEAL-2015-01[1]

Abstract

In the optimization of real-world activities the effects of solutions on related activities need to be considered. The use of isolated problem models that do not adequately consider related processes does not allow addressing system-wide consequences. However, sometimes the complexity of the real-world model and its interplay with related activities can be described by a combination of simple, existing, problems. In this work we aim to discuss strategies to combine existing algorithms for simple problems in order to solve a more complex master problem. New challenges arise in such an integrated optimization approach.

1 Introduction and Literature Review

The orienteering problem (OP) can be seen as a combination between the knapsack problem (KP) and the travelling salesperson problem (TSP) [1]. The traveling thief problem (TTP) is a similar combination of the TSP and the KP, but interleaves the two sub-problems to a higher degree [2]. Another problem is the knapsack constrained profitable tour problem (KCPTP) [3] that also combines a KP and a TSP.

In contrast to solving these problems with specialized algorithms or solution manipulation operators it is worthwhile to consider combining existing

[1]The original source of the publication is available at `https://doi.org/10.1007/978-3-319-27340-2_45`. The final publication is available at link.springer.com.

algorithms and studying the necessary interaction patterns that lead to good solutions in short time. Instead of solving the master, also denoted as integrated problem, several solvers are employed to obtain solutions to sub-problems. These solutions are called partial or sub-solutions to the master or integrated solution.

Past approaches that have to be mentioned include cooperative co-evolution [4] where the partitioning of problems has been used as a major tool. However, the problems in co-evolution are still tightly coupled within the variation loop of a genetic algorithm. The interaction between more specialized solvers for the sub-problems is not considered.

From a mathematical programming point, top-down approaches such as problem decomposition have to be mentioned. The application of Lagrangian decomposition on a mathematical formulation of the integrated problem allows splitting it into two sub-problems [5] [3]. Such a closed form mathematical formulation is not always possible however. We will perform a Lagrangian decomposition of the KCPTP next in order to derive an idea for a more general methodology.

1.1 Lagrange Decomposition

A description of the KCPTP model is given in equations (1) to (5).

$$\texttt{KCPTP: max} \sum_{i=1}^{n} y_i * p_i - t * \sum_{j=i+1}^{n} x_{ij} * d_{ij} \tag{1}$$

$$\sum_{i=1}^{n} y_i * w_i \leq K \tag{2}$$

$$\sum_{i=1}^{n} x_{ij} + \sum_{k=1}^{n} x_{jk} = 2 * y_j \quad \forall_j \in [1..n] \tag{3}$$

$$x \text{ has exactly one subtour} \tag{4}$$

$$y_1 = 1, y_i, x_{ij} \in 0, 1 \tag{5}$$

Constraint (2) is the knapsack constraint that limits the number of visited cities. By applying Langrangian decomposition y_i in equation (2) is substituted with a new decision variable z_i and a new equality constraint is added $y_i = z_i$ which is then again relaxed using Lagrangian multipliers λ. By rearranging the sums the objective function may be split with λ being a shared variable. The decomposed sub-problems are given in equations (6) to (12).

$$\texttt{KCPTP-LD-PTP}(\lambda):\quad \texttt{max}\sum_{i=1}^{n} y_i * (p_i - \lambda_i) - t * \sum_{j=i+1}^{n} x_{ij} * d_{ij} \tag{6}$$

$$\sum_{i=1}^{n} x_{ij} + \sum_{k=1}^{n} x_{jk} = 2 * y_j \quad \forall_j \in [1..n] \tag{7}$$

$$x \quad \text{has exactly one subtour} \tag{8}$$

$$y_1 = 1, y_i, x_{ij} \in 0, 1 \tag{9}$$

$$\texttt{KCPTP-LD-KP}(\lambda):\quad \texttt{max}\sum_{i=1}^{n} \lambda_i z_i \tag{10}$$

$$\sum_{i=1}^{n} z_i * w_i \leq K \tag{11}$$

$$z_1 = 1, z_i \in 0, 1 \tag{12}$$

The master problem in form of the Lagrange dual problem given in equations (13) to (14) then is to optimize λ such that the sum of the optimal solutions which is given by function ν to these two problems becomes minimal. This problem is piecewise linear and can be solved using a sub-gradient approach [5].

$$\texttt{KCPTP-LD: min } \nu(\texttt{KCPTP-LD-KP}(\lambda)) + \nu(\texttt{KCPTP-LD-PTP}(\lambda)) \tag{13}$$

$$\lambda_i \in \mathbb{R}^+ \tag{14}$$

2 A General Integrated Optimization Methodology

While Lagrange decomposition is a very useful concept, it becomes apparent that we cannot achieve a reduction to the TSP as we intended, but still have to solve the rather complex PTP, albeit without knapsack constraint. Furthermore, in Lagrangian decomposition the requirement is to solve sub-problems optimally in order to calculate the objective of the Lagrange dual problem. Combining this decomposition with heuristic approaches seems difficult or even impossible to achieve.

Still, we can use the hint that λ can be seen as a control parameter that adjusts the profits in the individual sub-problems. Generalizing this rather strict approach it seems feasible to assume that we may alter inputs such as profits so as to obtain solutions that are "good" with respect to the sub-problems *and* with respect to the master problem. In defining a rather general variegation strategy for sub-problem inputs we can come to a general methodology for solving integrated problems. Two slightly different approaches shall be presented first.

Sequential Approach: In the sequential approach sub-problems can be ordered such that the solution of one sub-problem describes restrictions to another problem. For example, in the KCPTP solutions to the KP limit the problem

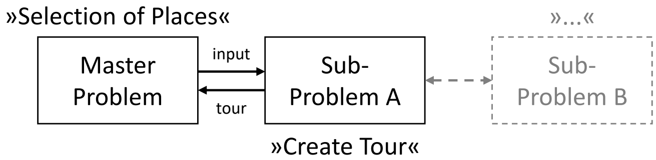

Figure 1: Sequential approach

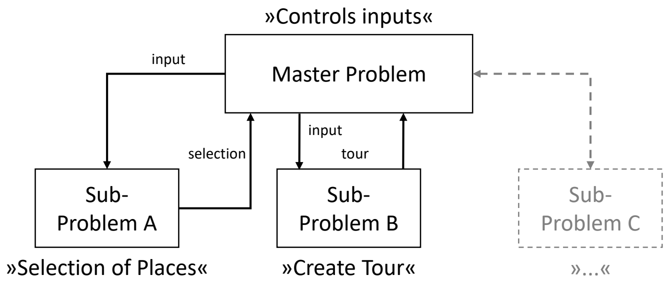

Figure 2: Cooperative approach

space of the TSP as only those customers need to be routed which have been selected. The integration between master and subordinate solver can be *tight* in that for every solution to the master problem a sub-problem needs to be solved or *loose* in that the master problem is solved and only for the best solution the problem reduction is performed and the subordinate solver is launched.

Cooperative Approach: In the cooperative approach, both problems are optimized concurrently, but the solvers are collaborating with each other. Collaboration could be *tight*, in that solvers exchange solutions during their search or *loose* in that the final results of solvers are used to parameterize the problems for a new optimization run. In the KCPTP, for example the solver for the KP can use a tour through all customers as a frame to evaluate the tour length on the subset of customers actually picked.

2.1 Orchestration of Solvers

As mentioned before for this approach to work well, it is required to control the subordinate solvers. For simple problems usually a number of methods are available such as problem-specific heuristics, metaheuristics, or exact approaches. But not in all cases is it useful to employ exact approaches or complex metaheuristics. It is not guaranteed that an optimal solution to a certain sub-problem instance can be integrated well into a solution to the master problem. The difficult part is to tune the sub-problem instances such that good solutions thereof align with good integrated solutions. Such a tuning can be thought of as an orchestration of solvers where two decisions have to be made.

A) *Solution Effort* - The effort with which the sub-problems are to be solved. For example, sub-problems that do not show to have a large impact on the

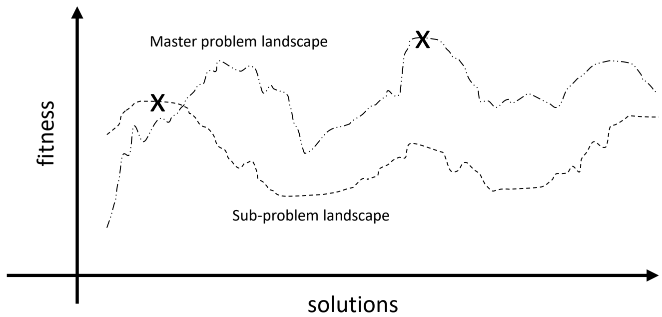

Figure 3: A simplified schematic drawing of the sub-problem landscape given the master problem's objective function with all other partial solutions fixed compared to the sub-problem's landscape. The example should illustrate that some local optima are well aligned, but the global optimum still leads to inferior solutions for the master problem.

master objective should be solved with less computational effort compared to sub-problems that have a very high impact.

B) *Sub-problem Variegation* - When the sub-problem landscape shows optima that are very bad for the integrated problem, the instance of the sub-problem needs to be altered in order to obtain solutions that are suitable for both master and sub-problem.

In this work we will not yet treat the first topic of deciding solution effort and use simple local search techniques to solve the sub-problems. But the second topic of variegating the sub-problem instances will be discussed in the following sections.

2.2 Sub-problem Input Variegation

The adjustment of the sub-problems' inputs can be seen as a change of the fitness landscapes with the goal to align it with the landscape of the master problem. Figure 3 attempts to sketch this idea schematically. A simple example shall further elaborate this idea: A remote, but profitable customer with low weight in a KCPTP is naturally attractive to be included in a knapsack solution. But on the other hand, servicing that customer requires to perform a larger detour. In a situation where travel costs are higher than the value of the detour an adjustment of this customer's profit will direct the knapsack solver to visit different customers.

The alignment between fitness landscapes can be seen as an optimization problem and has to be solved by a coordinating agent. In the case of Lagrangian decomposition such an "agent" would be the sub-gradient approach. However, in general for integrated problems sub-gradients are not available. Hence, a different measurement of the quality of such an input configuration is necessary.

The evaluation of some candidate input can be performed by applying a subordinate solver to the sub-problem and evaluate the obtained partial solution using the master problem's objective function (in combination with other partial solutions). This can then be seen as a fitness for the sub-problem's inputs. But while this approach is simple, the downside is that only the resulting solution, e.g. a local optimum, is used and the shapes of the landscapes are ignored. Another approach would be to compare a range of solutions obtained during the search from worse ones to better ones and calculate a correlation coefficient such as Spearman's rank correlation between sub-problem and master problem objective. The correlation coefficient can then be used as a measurement between the alignment of the two landscapes and therefore as a fitness for the actual sub-problem's inputs.

2.3 Algorithms

Algorithm 1 describes a loosely coupled sequential approach that includes the above mentioned input variegation. The coordinating agent requires four methods that *Reduce a problem* based on a solution as input, *Expand a solution* of a reduced problem to the original inputs, that *Evaluate* the master objective given the partial solutions and that *Variegate* the inputs of. Both reduction, expansion and variegation are only concerned with the type of the sub-problems and may be independent of whether the KCPTP, the OP, or the TTP is solved. *Evaluate* however is specific to the concrete master problem and contains the implementation of the master objective.

Algorithm 1 Pseudocode of a sequential integrated solver

```
procedure SOLVE(KpSolver, TspSolver)
    kp, kp′ ← InitializeKp()
    for iter = 0 To MaxIter do
        kpSol ← KpSolver.Solve(kp′)
        tsp ← Reduce(kpSol)
        tspSol ← Expand(TspSolver.Solve(tsp))
        masterObj ← Evaluate(kpSol, tspSol)
        kp′ ← Variegate(kp, masterObj)
    end for
end procedure
```

3 Results

All results are averaged over 10 runs and were calculated on a laptop equipped with a 3^{rd} generation Intel Core i7 running at 2.6 Ghz. Only a single core was used in all tests. Results are computed for some instances of the KCPTP, OP, and the TTP. For the KCPTP we used benchmark instances for orienteering problems that originally only specified a profit per location [6, 7]. The weights for the knapsack were generated to be correlated to the profits using equation (15) where $U(0, 1)$ returns a random number in the interval $[0, 1)$. The maximum size of the knapsack has been calculated using equation (16). The transport cost factor for each instance has been scaled in a geometric progression between minimum and maximum profit to distance ratio for a total of 6 different variants per instance and results have been averaged. For the TTP, we relaxed the constraints of having to visit all locations and visit only the locations where something is actually stolen.

$$w_i = T_{max} * (0.1 + U(0,1) * 0.8 * \frac{p_i - \texttt{min}(p)}{\texttt{max}(p) - \texttt{min}(p)}) \tag{15}$$

$$K = (0.2 + U(0,1) * 0.6) * \sum_i^n w_i \tag{16}$$

Table 1: Results of the sequential approach applied to several instances of the KCPTP and TTP problems.

	With Variegation			Without Variegation		
KCPTP Instances	Avg	StdDev	[sec]	Avg	StdDev	[sec]
chao64.set_64_1_25	**701.1**	9.7	1.3	698.7	7.2	0.7
chao64.set_64_1_50	**856.2**	9.2	1.5	849.5	6.3	1.0
chao64.set_64_1_80	**801.7**	6.9	1.5	799.1	5.2	0.9
tsi1.budget_15	**11.2**	5.1	0.4	8.6	4.2	0.1
tsi1.budget_30	**16.2**	5.1	0.4	11.5	4.1	0.1
tsi1.budget_60	**39.3**	4.4	0.5	30.6	3.1	0.3
OP Instances	Avg	StdDev	[sec]	Avg	StdDev	[sec]
chao64.set_64_1_25	**109.2**	22.9	5.0	-	-	0.5
chao64.set_64_1_50	382.2	37.4	0.2	**445.2**	10.9	0.2
chao64.set_64_1_80	**501.6**	11.4	0.4	492.6	9.6	0.3
tsi1.budget_15	-	-	1.1	-	-	0.2
tsi1.budget_30	**57.0**	11.8	0.7	-	-	0.2
tsi1.budget_60	141.5	21.4	0.1	**175.5**	2.8	0.2
TTP Instances berlin52_n51	Avg	StdDev	[sec]	Avg	StdDev	[sec]
bounded-strongly-corr_01-10	**15269.5**	529.7	1.0	14246.9	505.3	0.6
uncorr-similar-weights_01-10	**7934.2**	269.6	1.0	5476.9	329.7	0.5
uncorr_01-10	**8777.0**	405.3	1.0	5559.1	422.5	0.5

The results in Table 1 show that the variegation of input is beneficial for the search when using the same algorithm. For the OP, also the knapsack weights were variegated in order to achieve more feasible solutions. Still, constraint variegation still needs to be improved as the results indicate. The difference in runtime can be explained by the variegation strategy that employed a CMA-ES algorithm to modify the profit vector. Both times the same number of iterations were given. While the results for the KCPTP do not convince as much, the results for the TTP show that input variegation actually makes a significant difference and that an adjustment of the sub-problem's fitness landscapes is necessary to deliver better results for the master problem.

4 Conclusions and Future Work

In this work we discussed a general methodology for solving integrated problems. We showed that complex problems that consist of simpler sub-problems can be decomposed using Lagrange decomposition. We also discussed the disadvantage of these techniques in the requirement of exact solution approaches and went on to present a more general methodology for solving integrated problems.

Thoughts have been presented on how a variegation of the sub-problem's inputs is able to shift the sub-problem's fitness landscape so that it aligns with the fitness landscape of the master problem. This more general methodology may be applied between arbitrary combinations of sub-problems and is not limited to closed-form mathematical descriptions. First results have shown that the variegation is indeed beneficial for the search. The acceptance of this approach will certainly depend on the simplicity of its use. But to be successful, it must be considerably simpler than the development of a new algorithm for solving a new complex problem while still achieving competitive results.

Acknowledgments

The work described in this paper was done within the COMET Project Heuristic Optimization in Production and Logistics (HOPL), #843532 funded by the Austrian Research Promotion Agency (FFG).

References

[1] Vansteenwegen, P., Souffriau, W., Oudheusden, D.V.: The orienteering problem: A survey. European Journal of Operational Research **209** (2011) 1–10

[2] Polyakovskiy, S., Bonyadi, M.R., Wagner, M., Michalewicz, Z., Neumann, F.: A comprehensive benchmark set and heuristics for the traveling thief problem. In: Genetic and Evolutionary Computation Conference, GECCO '14, Vancouver, BC, Canada (2014) 477–484

[3] Feillet, D., Dejax, P., Gendreau, M.: Traveling salesman problems with profits: An overview. Transportation Science **39** (2001) 188–205

[4] Potter, M.A., Jong, K.A.D.: A cooperative coevolutionary approach to function optimization. In: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature. PPSN III, London, UK, UK, Springer-Verlag (1994) 249–257

[5] Pirkwieser, S., Raidl, G., Puchinger, J.: Combining lagrangian decomposition with an evolutionary algorithm for the knapsack constrained maximum spanning tree problem. In Cotta, C., van Hemert, J., eds.: Evolutionary Computation in Combinatorial Optimization. Volume 4446 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2007) 176–187

[6] Chao, I.M., Golden, B.L., Wasil, E.A.: A fast and effective heuristic for the orienteering problem. European Journal of Operational Research **88** (1996) 475–489

[7] Tsiligirides, T.: Heuristic methods applied to orienteering. Journal of the Operational Research Society **35** (1984) 797–809