A Population-Based Local Search for Solving a Bi-objective Vehicle Routing Problem

Joseph M. Pasia[1,2], Karl F. Doerner[2], Richard F. Hartl[2], and Marc Reimann[3]

[1] Department of Mathematics, University of the Philippines-Diliman, Quezon City, Philippines
jmpasia@up.edu.ph
[2] Department of Management Science, University of Vienna, Vienna, Austria
{richard.hartl, karl.doerner}@univie.ac.at
[3] Institute for Operations Research, ETH Zurich, Zurich, Switzerland
marc.reimann@ifor.math.ethz.ch

Abstract. In this paper we present a population-based local search for solving a bi-objective vehicle routing problem. The objectives of the problem are minimization of the tour length and balancing the routes. The algorithm repeatedly generates a pool of good initial solutions by using a randomized savings algorithm followed by local search. The local search uses three neighborhood structures and evaluates the fitness of candidate solutions using dominance relation. Several test instances are used to assess the performance of the new approach. Computational results show that the population-based local search outperforms the best known algorithm for this problem.

1 Introduction

The vehicle routing problem (VRP), introduced in [1], consists of finding the optimal route for a fleet of vehicles, starting and ending at a single depot, that must serve a set of n customer demands such that each customer is visited by only one vehicle route. If each vehicle can only collect a maximum capacity of Q units of demands, then the problem is known as capacitated VRP (CVRP).

Although exact approaches [2,3] have been proposed to solve the VRP, many approximation methods have been developed since VRP has been proven $\mathcal{N}$P-hard [4]. Some examples of these methods are the classical heuristics such as the well-known savings algorithm [5]. These methods put more emphasis on the ability to search a good feasible solution.

Over the last 15 years, increasing research effort has been devoted on the development of metaheuristic approaches since they can search the solution space much more thoroughly. The metaheuristic approaches use either the principle of local search or population search [6,7]. Local search methods approximate a region of the Pareto front in the direction given by a weight vector λ using a single solution. The aggregation of the objectives via λ is usually used in order

C. Cotta and J. van Hemert (Eds.): EvoCOP 2007, LNCS 4446, pp. 166–175, 2007.

to focus the search on the region of interest. On the other hand, population searches such as the adaptive memory procedure [8] and SavingsAnts [9] maintain a pool of solutions called population that move together through evolution process towards the Pareto front. In many cases, local search methods are used to improve the quality of the solutions in the population.

In recent years, metaheuristics have also been used to solve vehicle routing problems with multiple objectives. For instance, a bi-objective CVRP with route balancing (CVRPRB) was tackled using genetic algorithms [10,11]. The two objectives of CVRPRB are (i) to minimize the sum of the total distance travelled by each vehicle and (ii) to minimize the difference between the longest and shortest vehicle tours.

In general, there is no single solution that simultaneously accomplishes the objectives of a bi-objective optimization problem. Hence, the Pareto optimal solutions or sometimes called the set of efficient solutions are considered. We say that a solution x is an efficient solution if there exists no other feasible solution y such that $f_k(y) \leq f_k(x),$ for $k = 1, 2$ and $f_k(y) < f_k(x)$ for some k. Otherwise, we say that x is dominated by y and we denote this by $y \prec x$. In addition, if x^* is Pareto optimal then $z^* = f(x^*)$ is called nondominated vector and the set of all nondominated vectors is referred to as nondominated frontier (or Pareto front or trade-off surface).

In this paper, we propose another approach in solving the CVRPRB by combining the principles of population search and local search. Our approach starts by generating a pool of good starting solutions using a randomized savings algorithm. These solutions then undergo local search that uses three neighborhood structures. The candidate solutions in the local search are evaluated using dominance relation. In the following discussions, we will refer to our method as population-based local search or P-LS.

This paper is organized as follows: Section 2 explains the details of P-LS. Section 3 discusses the numerical results of the study and Section 4 provides a short conclusion of the study.

2 Population-Based Local Search for CVRPRB

The P-LS method follows the two basic steps given by Algorithm 1. The first step is the initialization phase or the creation of the starting solutions. The second step is the local search phase where we apply our local search operators. In the following sections, we describe the details how we implemented these steps.

Algorithm 1. Basic algorithmic framework of P-LS

While (condition is satisfied) Do /*`We call this loop as generation`*/
  Create a pool of solutions $\mathcal{S}$
  Apply local search on the solutions of $\mathcal{S}$

2.1 Initialization Phase of P-LS

Being a population-based heuristic, the starting solutions of P-LS are important. It was demonstrated in [12,13,14] that spending time in creating a good initial population improves the convergence in optimization. Hence, we use a pool of good starting solutions for our P-LS. Our strategy in creating the initial solutions uses the randomized savings algorithm or savings algorithm with candidate list. The savings algorithm starts with the assignment of each customer to a separate tour. The customers or the partial tours are then combined based on the savings values given by $s(i, j) = d(i, 0) + d(j, 0) - d(i, j)$ where $d(i, j)$ is the distance between customers i and j and the index 0 denotes the depot. Clearly, $s(i, j)$ is the value saved when i and j are combined instead of serving them by two different tours. The combination of customers begins with the largest savings until no more combination is feasible.

Instead of combining the customers having the largest savings, the randomized savings algorithm creates a candidate list C of feasible combinations of customers i and j. The set C consists of the combinations with the $|C|$ best savings values. Each combination in the candidate list is selected with equal probability. After combining the selected pair of customers, the combinations in C that become infeasible are removed and we maintain the size of C by adding the next feasible combinations with the best savings values.

To generate the starting solutions of P-LS, a pool $\mathcal{S}$ of identical solutions are initially created. These identical solutions assign each customer to a separate tour. We improve each of these solutions by combining the customers based on the randomized savings algorithm. We then apply 2-opt local operator to the solutions of $\mathcal{S}$ in order to avoid artificially balanced solutions [11]. In P-LS, we apply our local search operators only to the nondominated solutions. Hence, we remove all dominated solutions in $\mathcal{S}$ before proceeding to the local search phase.

2.2 Local Search Phase of P-LS

The local search phase uses one intratour-neighborhood called 2-opt and two intertour-neighborhood structures namely move and swap. The move neigborhood inserts the customer of one partial tour to another partial tour. On the other hand, the swap neighborhood exchanges two customers from different partial tours. After performing either the insertion or exchange operator, the 2-opt operator is applied to the affected partial tours.

The basic step of the local search phase is described in Algorithm 2. This step was called multiobjective local search in [15] and pareto local search in [16] and it was implemented in scheduling problem using a single neighborhood structure. The main difference of our approach compared to the existing techniques is that we use three different neighborhood structures. It has been shown in [9,17] that these three neighborhoods are effective for the single objective classical VRP.

Starting from a solution $z \in \mathcal{S}$, the feasible solutions of the first neighborhood $\mathcal{N}_1$ of z are explored. All feasible neighboring solutions are compared and the dominated ones are removed. Each of the remaining efficient solution will undergo the same process as z i.e., its entire neighborhood is searched and all the dominated solutions are removed. We repeat the entire process of searching the whole neighborhood and removing the dominated solutions until all solutions in the neighborhood are dominated. When this happen, Algorithm 2 is repeated on $\mathcal{P}$ using the second neighborhood $\mathcal{N}_2$. Unless the termination conditions are not satisfied after performing $\mathcal{N}_2$, we apply $\mathcal{N}_1$ again on the pareto set returned by $\mathcal{N}_2$. Since exploring the entire neighborhood of a given solution is computationally expensive, we only allow a certain number of solutions to undergo the local search. Figure 1 illustrates this process.

Algorithm 2. Basic step of the local search phase

Pareto set $\mathcal{S}$
t = 0
While ($\mathcal{S}$ is non-empty and t $< M$) Do /*`We call this loop as iteration`*/
  Pareto set $\mathcal{C} = \emptyset$
  Forall $z \in \mathcal{S}$ do
    Pareto set $\mathcal{L} \leftarrow z$
    Forall $w \in \mathcal{N}_i(z)$ do
      If $w \notin \mathcal{L}$ and w is not dominated by $\mathcal{L}$ then
      Set $\mathcal{L}$ = nondominated solutions of $(\mathcal{L} \cup \{w\})$ and
      Set $\mathcal{C}$ = nondominated solutions of $(\mathcal{C} \cup \{w\})$
  $\mathcal{S} = \mathcal{C}$
  Update Pareto set $\mathcal{P}$ by $\mathcal{C}$
  t = t + 1
Return $\mathcal{P}$

3 Numerical Results

The numerical analysis was performed on a set of benchmarks described in [18]. The set of benchmarks consists of 7 test instances having 50 to 199 customers and a single depot. Four of the instances were generated such that the customers are uniformly distributed on a map and the remaining instances feature clusters of customer locations. All test instances have capacity constraints. We performed all our methods on a personal computer with 3.2 Ghz processor; the algorithms were coded in C++ and compiled using GCC 4.1.0 compiler.

3.1 Evaluation Metrics

The use of unary quality indicators has become one of the standard approaches in assessing the performance of different algorithms for bi-objective problems. It complements the traditional approach of using graphical visualization which may provide information on how the algorithm works [19]. This study considered three unary quantitative measures namely, the *hypervolume indicator*, *unary epsilon indicator*, and *R3 indicator*.

Th hypervolume indicator (I_H) measures the hypervolume of the objective space that is weakly dominated by an approximation set or the set containing the

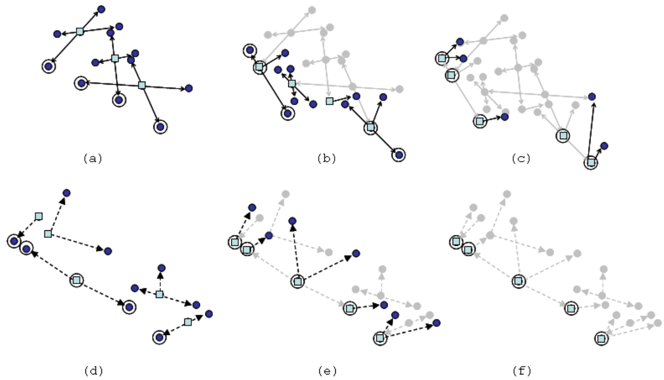

Fig. 1. The boxes represent the efficient solutions. The solid circles are the feasible solutions in the neighborhood of an efficient solution. The encircled solutions are the efficient neighbors which become the new efficient solutions of $\mathcal{S}$. (a) Four efficient solutions are found using $\mathcal{N}_1$. (b) After exploring the neighborhoods of these efficient solutions, 3 new efficient solutions are found and 2 efficient solutions from (a) remain efficient. (c) Not a single neighbor of efficient solutions in (b) is efficient. (d) $\mathcal{N}_2$ is then used and 4 of the neighbors are found to be efficient. (e) The $\mathcal{N}_2$-neighborhood of the efficient solutions in (d) does not have any efficient solution this time. (f) Since $\mathcal{N}_2$ generated new efficient solutions, the current set of efficient solutions will again be explored using $\mathcal{N}_1$.

nondominated frontier of an approximation method [20]. This is calculated using a boundary point that is dominated by all approximation sets. It has a desirable property that whenever an approximation set A is better than approximation set B, then the hypervolume of A is greater than B.

The unary epsilon indicator I_ϵ gives the minimum factor ϵ such that if every point in reference set X is multiplied by ϵ, then the resulting approximation set is weakly dominated by A. For minimization problem, this indicator is formally defined by:

$$I_\epsilon(A) = I_\epsilon(A, X) = \inf_{\epsilon \in \mathrm{X}} \left\{ \forall z^2 \in X \exists z^1 \in A : z^1 \preceq_\epsilon z^2 \right\} \quad (1)$$

where the ϵ-dominance relation is defined as $z^1 \preceq_\epsilon z^2 \Leftrightarrow \forall i \in 1, 2, \ldots, n : z_i^1 \leq \epsilon \cdot z_i^2$. Note that a small ϵ value is preferable.

The R3 indicator (I_{R3}) used in this study is one of R indicators proposed in [21]. Given a set of weight vectors Λ, this indicator is defined as:

$$I_{R3}(A) = I_{R3}(A, X) = \frac{\sum_{\lambda \in \Lambda} \left[u^*(\lambda, X) - u^*(\lambda, A) \right] / u^*(\lambda, A)}{|\Lambda|} \quad (2)$$

where u^* is the maximum value attained by a utility function u_λ with weight λ, i.e., $u^*(\lambda, A) = \max_{z \in A} u_\lambda(z)$. In this study, the utility function is given by:

$$u_\lambda(z) = -\left(\max_{j \in 1..n} \lambda_j |z_j^* - z_j| + \rho \cdot \sum_{j \in 1..n} |z_j^* - z_j| \right) \tag{3}$$

where z^* is the ideal point and ρ is a sufficiently small positive real number. The values of I_{R3} range from -1 to 1 where values close to -1 are superior.

3.2 Parameter Settings

The initial number of solutions in each pool $\mathcal{S}$ is equal to the number of customers. The size of the candidate list is given by $\lfloor 0.10 \times (\# \text{ of customers}+1) \rfloor$. The maximum number of solutions in $\mathcal{S}$ that undergo local search is ten. These 10 solutions are chosen so that the corresponding nondominated points are evenly distributed in the objective space i.e., dividing the relevant section in equally-sized segments. The value of M is either 100 or 150 for the move-neighborhod and 50 to 75 for the swap-neighborhood.

3.3 Analysis

Ten runs with different random seeds were performed for each of the test instance. Before applying the different unary indicators, all approximation sets are normalized between 1 and 2. The boundary point used in the hypervolume indicator is (2.1, 2.1), and the ρ and $|\Lambda|$ in I_{R3} are 0.01 and 500 respectively [19]. The reference set for each test instance consists of the points that are not dominated by any of the approximation sets generated by all algorithms under consideration.

The performance of P-LS is compared to the Nondominated Sorting Genetic Algorithm II (NSGA II) developed in [22] and proposed as algorithm for CVRPRB in [10]. The NSGA II approach for CVRPRB was enhanced by parallelization and by the use of elitist diversification mechanism. Several implementations of NSGA II which have different number of Power4 1.1 Ghz processors were examined. The average computation time ranged from 900 to 5200 seconds. We used the only available results of one of the NSGA II variants found in `http://www2.lifl.fr/jozef/results_VRPRB.html` for comparison.

In this study, we demonstrate the importance of randomized savings algorithm in generating the pools of initial solutions. To do this, we compare our P-LS to a P-LS that generates solutions randomly. We call this method as P-LS-0. In addition, we also investigate the advantage of allowing our P-LS to use more starting solutions by comparing our method to a P-LS where M=∞. We refer to this method as P-LS-1. All three methods are terminated after reaching a computational time that ranges from 600 to 30000 seconds depending on the test instance.

Figure 2 shows the boxplots of the different unary indicators for all test instances[1]. A boxplot provides a graphical summary of the median, the range and

[1] The first number in the name of each test instance corresponds to the total number of customers and depot. The last letter determines whether the customers are clustered (c) or uniformly distributed (e).

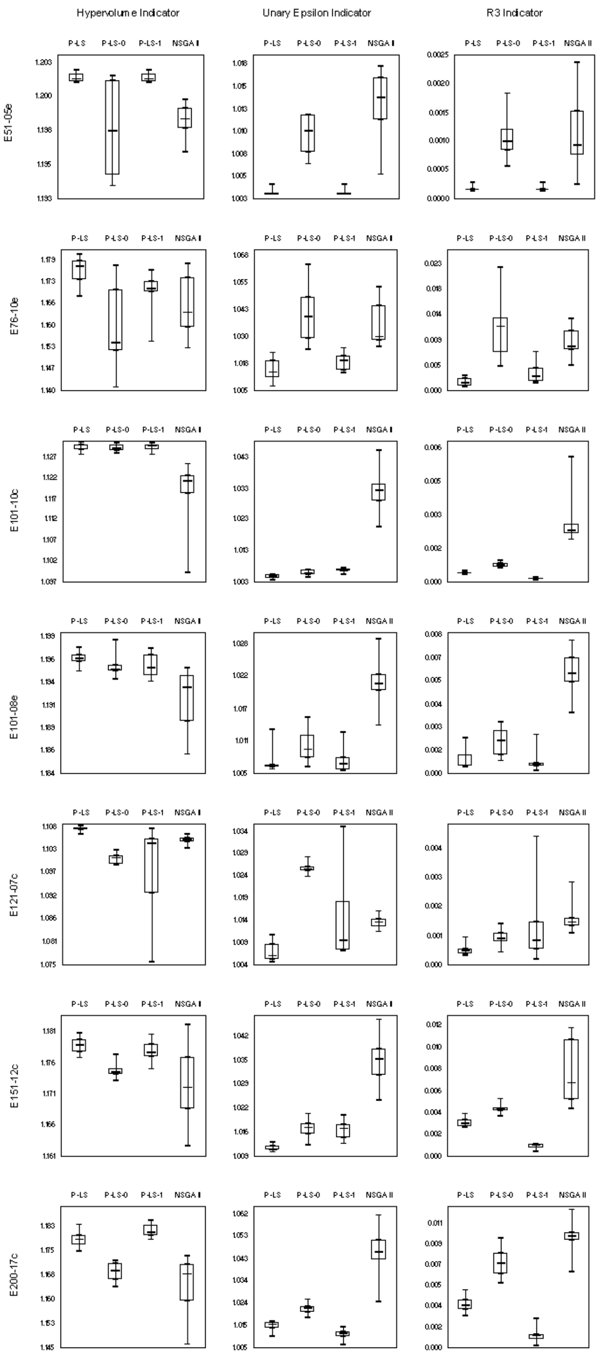

Fig. 2. Box plots of the different unary indicators for all test instances. High hypervolume values and low unary epsilon and R3 values are preferable.

inter-quartile range, and the orientation of the median relative to the quartiles for a set of data.

P-LS and P-LS-0. For all test instances, the boxplots of P-LS are better than that of P-LS-0. It can be observed that the positive effect of the randomized savings algorithm is more evident in larger instances (at least 120 customers). Thus the good performance of P-LS can be attributed to the quality of starting solutions generated by the randomized savings algorithm.

P-LS and P-LS-1. Allowing the P-LS to have more pools improves the performance P-LS with respect to hypervolume and unary epsilon indicators in all but one test instance. However, the R3 indicator gets worse in many test instances. To explain why this may happen, we consider the first runs of P-LS and P-LS-1 for test instance E151-12c. Figure 3 provides the unary quality indicators and the plots of the efficient points for this case. It is clear from these plots that the best objective values of P-LS with respect to total distance are much better than that of P-LS-1 and this translates into bigger hypervolume. In addition, P-LS-1 is slightly better than P-LS in the middle region. It has more points in this region and most of them belong to the reference set. This slight difference in the middle area may not compensate the gains of P-LS with respect to hypervolume but this helps P-LS-1 to have a better R3 indicator value. Regarding the unary epsilon, it is also clear from the graph that P-LS-1 requires a bigger factor in order for its approximation set to weakly dominate the reference set.

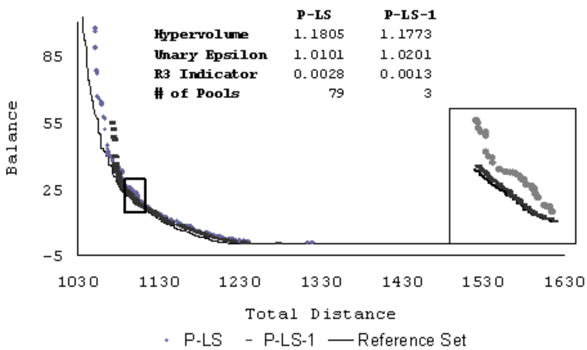

Fig. 3. Plots the nondominated frontiers of the first run of P-LS and P-LS-1 for test instance E151-12c. The region in the middle enclosed in a box is enlarged.

P-LS and NSGA II. The boxplots of the three unary quality indicators of P-LS is better than NSGA II for all instances. In fact, all the median values of P-LS are better than the median values of NSGA II and for many cases, the median values are better than the best values of NSGA II. This shows that our P-LS approach performs well in all the test instances that were used. It is also interesting to note that P-LS-0 and P-LS-1 are also better than NSGA II with respect to the three unary quality indicators for some instances. For example, all three P-LS approaches are better than NSGA II in test instances E101-10c and E101-08e.

4 Conclusion

In this study, we proposed the P-LS or population-based local search to solve a bi-objective vehicle routing problem. The P-LS uses a pool of good starting solutions generated by a randomized savings algorithm. All efficient solutions in the pool were allowed to undergo local search. Our local search approach uses three neighborhoods and and the candidate solutions were evaluated using dominance relation.

Computational results showed that our P-LS approach performed well compared to NSGA II with respect to three unary quality indicators. We have also demonstrated that using the randomized savings algorithm improves the solutions quality of P-LS. We have also shown that the performance of P-LS improves when we allow our P-LS to use more starting pools of initial solutions.

In the future, we intend to apply our P-LS to the large-scale VRP instances. We also intend to extend our P-LS approach to VRP with more than two objectives, e.g. load balancing, number of vehicles. Moreover, we plan to apply P-LS to other multiobjective combinatorial problems.

References

1. Dantzig, G., Ramsey, J.: The truck dispatching problem. Management Science **6** (1959) 80–91
2. Baldacci, R., Hadjiconstantinou, Mingozzi, A.: An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. Operations Research **52**(5) (2004) 723–738
3. Fukasawa, R., Longo, H., Lysgaard, J., Poggi de Aragão, M., Reis, M., Uchoa, E., Werneck, R.: Robust branch-and-cut-and-price for the capacitated vehicle routing problem. Mathematical Programming **106**(3) (2006) 491–511
4. Lenstra, J., Kan, A.: Complexity of vehicle routing and scheduling problem. Networks **11** (1981) 221–227
5. Clarke, G., Wright, J.: Scheduling of vehicles from a central depot to a number of delivery points. Operations Research **12** (1964) 568–581
6. Cordeau, J., Gendreau, M., Laporte, G., Potvin, J., Semet, F.: A guide to vehicle routing heuristics. Journal of the Operational Research Society **53** (2002) 512–522
7. Ehrgott, M., Gandibleux, X.: A survey and annotated bibliography of multiobjective combinatorial optimization. OR Spektrum **22** (2000) 425–460
8. Rochat, Y., Taillard, E.: Probabilistic diversification and intensification in local search for vehicle routing. Journal of Heuristics **1** (1995) 147–167
9. Doerner, K., Gronalt, M., Hartl, R.F., Reimann, M., Strauss, C., Stummer, M.: SavingsAnts for the vehicle routing problem. In Cagnoni, S., et al., eds.: Proceedings of Applications of Evolutionary Computing : EvoWorkshops 2002. Volume 2279 of LNCS. (2002) 11–20

10. Jozefowiez, N., Semet, F., Talbi, E.G.: Enhancements of NSGA-II and its application to the vehicle routing problem with route balancing. In Talbi, E., et al., eds.: Proceedings of 7th International Conference Artificial Evolution-EA 2005. Number 3871 in LNCS, Springer-Verlag (2006) 131–142
11. Jozefowiez, N., Semet, F., Talbi, E.G.: Parallel and hybrid models for multi-objective optimization: Application to the vehicle routing problem. In Guervós, J., et al., eds.: Parallel Problem Solving from Nature—PPSN VII. Number 2439 in LNCS, Granada, Spain, Springer-Verlag (2002) 271–280
12. Haubelt, C., Gamenik, J., Teich, J.: Initial population construction for convergence improvement of moeas. In Coello, C., Aguirre, A., Zitzler, E., eds.: Evolutionary Multi-criterion Optimization, EMO'2005. Volume 3410 of LNCS. (2005) 191–205
13. Gandibleux, X., Morita, H., Katoh, N.: The supported solutions used as a genetic information in a population heuristic. In: Evolutionary Multi-criterion Optimization, EMO'2001. Volume 1993 of LNCS., Zurich, Switzerland (2001) 429–442
14. Morita, H., Gandibleux, X., Katoh, N.: Experimental feedback on biobjective permutation scheduling problems solved with a population heuristic. Foundations of Computing and Decision Sciences **26**(1) (2001) 23–50
15. Arroyo, J., Armentano, V.: Genetic local search for multi-objective flowshop scheduling problems. European Journal of Operational Research **167** (2005) 717–738
16. Basseur, M., Seynhaeve, F., Talbi, E.: Path relinking in pareto multi-objective genetic algorithms. In Coello, C., Aguirre, A., Zitzler, E., eds.: Evolutionary Multi-criterion Optimization, EMO'2005. Volume 3410 of LNCS. (2005) 120–134
17. Reimann, M., Doerner, K., Hartl, R.: D-ants: Savings based ants divide and conquer the vehicle routing problem. Computers & Operations Research **31**(4) (2004) 563–591
18. Christofides, N., Mingozzi, A., Toth, P.: The vehicle routing problem. In Christofides, N., Mingozzi, A., Toth, P., Sandi, C., eds.: Cominbatorial Optimization. John Wiley and Sons (1979)
19. Knowles, J., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical Report TIK-Report No. 214, Computer Engineering and Networks Laboratory, ETH Zurich, Gloriastrasse 35, ETH-Zentrum, 8092 Zurich, Switzerland (2006)
20. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans. Evolutionary Computation **3**(4) (1999) 257–271
21. Hansen, M., Jaszkiewicz, A.: Evaluating the quality of approximations to the non-dominated set. Technical Report Technical Report IMM-REP-1998-7, Technical University of Denmark (1998)
22. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II. In: PPSN. (2000) 849–858