Illustrative example for the paper titled "A new algorithm for resource-constrained project scheduling with breadth and depth of skills" by Jakob Snauwaert & Mario Vanhoucke.

Example of problem and solution approach

In this online extension, we present a complete example of the MSRCPSP with breadth and depth and our solution procedure (see figure 1).

Problem: At the top of figure 1, the problem details are presented. In this example, there are 5 activities, 4 resources and 2 skills. The project network displays the finish-start precedence relations between the activities. The activity characteristics show the skill requirements per activity r_{ij} and the standard processing time p_i. The resource characteristics represent the depth d_{jk} and resource criticality RC_k^a of each resource. The skill criticality SC_j and ranked skill criticality RSC_j are shown in the skill characteristics table. The RC_k^a and SC_j are calculated using equations (27) and (24). The RSC_j of activities 1 and 2 is, respectively, equal to 1 and 2 because $SC_1 < SC_2$.

Representation: In the representation, both the activity list and the priority list are displayed as explained in section 4.2.1. The solution procedure contains, in this case, three priority rules, PR_1, PR_2 and PR_3.

Crossovers: To generate new solutions, two crossovers are explained in section 4.2.4. For the low demand range crossover (for activity lists), ranges with a q_i lower than $\overline{q_i}$ (shown in light grey in figure 1) are determined. In this case, two ranges are selected, of length 1 and 3. These ranges are then split by the crossover to create the child solution. Activity 2 and 5, which are situated outside the crossover points, are copied from parent 1 in the child solution. The other activities are added to the child solution in the order of parent 2. The random two-point crossover (for priority lists), operates in a similar manner, the only difference being that the crossover points are chosen randomly.

Mutators: The swap mutator swaps activities 1 and 2 of the AL to create a new solution. The modify mutator changes the priority rule for activity 2, from PR_1 to PR_2.

PSGS: The PSGS generates a schedule and resource assignment from the AL and the PL shown in that section of the example. In this case, activity 1 is scheduled first, at $t = 0$, and resources 1, 2 and 3 are assigned to it, according to PR_1. The average depth of the skills assigned to activity 1 is equal to 1.25, so $p_1^a = 5 \cdot \frac{1}{1.25} = 4$. Before incrementing t, activity 2 can be performed by resource 4 starting at $t = 0$. Since, the depth of resource 4 is equal to 1, $p_2^a = 10$. Next, we move to $t = 4$, at which time activity 1 finishes. Next, activity 3 is scheduled and resources 1 and 2 are assigned by using priority rule PR_3. Their average depth is equal to 1, so $p_3^a = p_3$. Then the time is incremented to the finishing time of activity 3, $t = 9$. After which, activity 4 is scheduled with resource 2 performing skill 2 specified by PR_2. Because $d_{22} = 1.25$, $p_4^a = 8$. Finally, we move to $t = 10$, activity 2

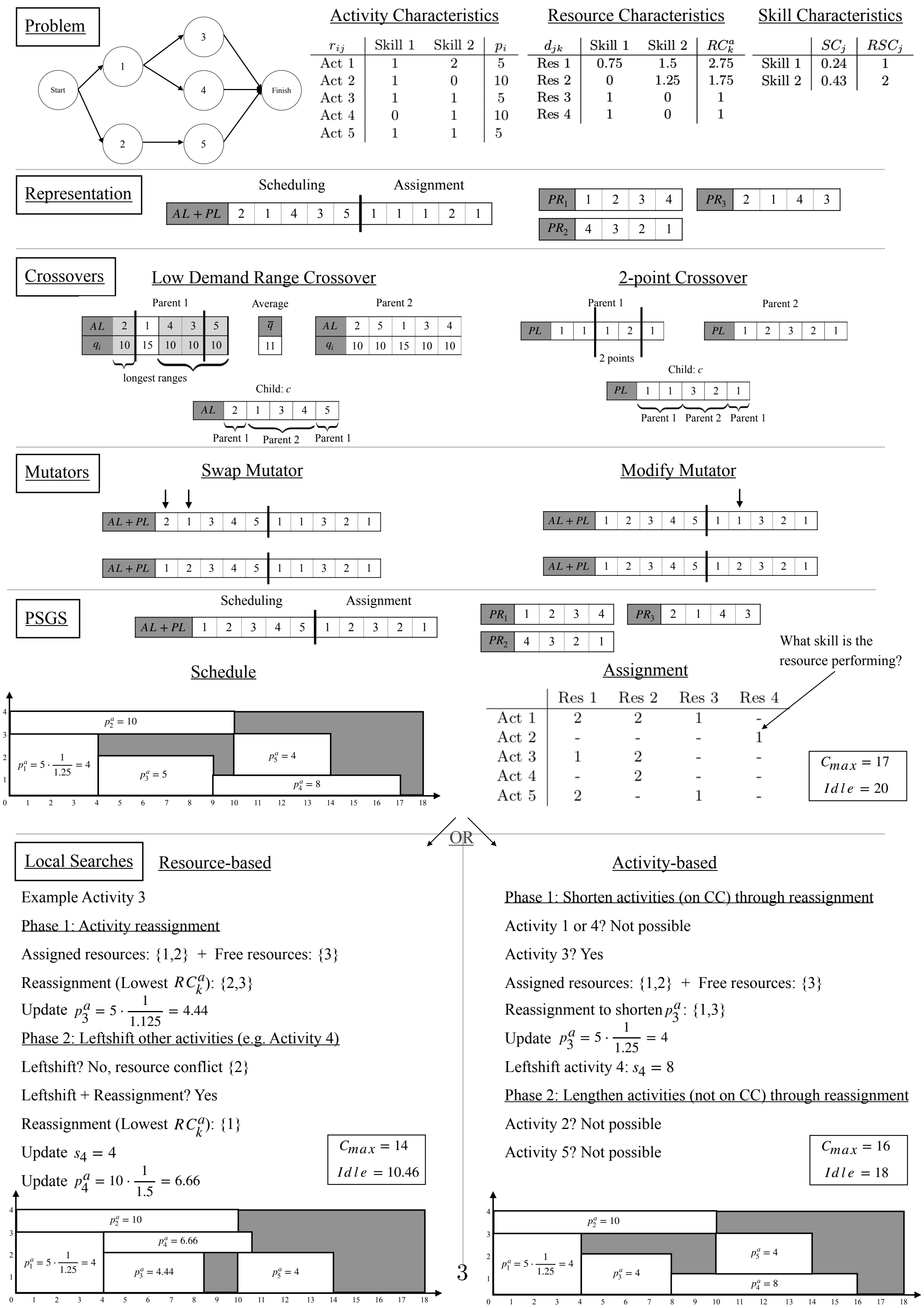

Figure 1: Small example of the problem and the different parts of the solution approach

finishes and activity 5 is started. Using PR_1 resources 1 and 3 are assigned and $p_5^a = 4$. All activities are scheduled and we arrive at a makespan (C_{max}) equal to 17 and an idle time ($Idle$) equal to 20.

Local Searches: The created schedule is improved by either the resource-based local search (RLS) or the activity-based local search (ALS).

In this example, we only perform the RLS on activity 3. In phase 1 of the RLS, the resources assigned to activity 3 (A_3^r) and the free resources (U_3^r) are used to do a reassignment based on the lowest resource criticality RC_k^a. Due to this, skill 1 required by activity 3 is now performed by resource 3 and resource 1 becomes available. Since $d_{22} = 1.25$ and $d_{31} = 1$, the average assigned depth is equal to 1.125 and $p_3^a = 4.44$. In phase 2, the goal is to profit from the new set of available resources. First, we check if activity 4 can be shifted to the left without a reassignment, which is impossible because both activities 3 and 4 are performed by resource 2. Next, we check if we can leftshift activity 4 with a reassignment. Seeing that activity 4 can be performed by the only free resource (resource 1), we can schedule it at $t = 4$. The adjusted processing time of activity 4 decreases to 6.66. The changes made by performing the RLS on activity 3 reduce the makespan to 14 and the idle time to 10.46.

The ALS checks in phase 1 if it can shorten the p_i^a of any critical activities. In this example, activities 1 and 4 already have the shortest possible adjusted processing time in the current schedule. Only activity 3 can be shortened by assigning resources 1 and 3 to, respectively, skill 2 and skill 1. The p_3^a is updated to 4, because of the increased assigned depth. Due to this reassignment, activity 4 can now be shifted to the left with 1 time unit. In phase 2 the ALS tries to increase the duration of not critical activities in order to decrease the idle time. No such improvement is possible in this case. The changes made in the ALS improve the makespan and idle time to 16 and 18.