Problem planowania projektów z wieloma wymaganymi umiejętnościami i ograniczonymi zasobami został zdefiniowany przez naukowców związanych z Politechniką Wrocławską, Ich celem było stworzenie problemu, która miał by jednoczesny balans prostoty implementacji i wierności rzeczywistej sytuacji.

Celem klasycznego problemu jest przydzielenie zasobów do zadań w taki sposób aby zminimalizować czas i/lub koszt wykonania projektu. Jednakże pojawiają się w nim także ograniczenia. Pierwszym z nim jest fakt, że niektóre zadania, mogą do rozpoczęcia pracy nad nimi wymagać zakończenia innych zadań. A drugim, że dany zasób może zostać przydzielony tylko do jednego zadania naraz i musi je skończyć w całości, bez możliwości dzielenia pracy pomiędzy dwoma rozpoczętymi zadaniami.

Problem ten został rozszerzony tutaj przez dodanie umiejętności. Każdy zasób posiada pewien zbiór umiejętności na pewnym poziomie, a każde zadanie ma określone jedno wymaganie. Dane zadanie może mieć przydzielony dany zasób, tylko jeżeli posiada on daną umiejętność na równym lub wyższym poziomie niż to potrzebne.

Problem ten został opracowany razem ze współpracy z inżynierami z firmy Volvo. Dzięki temu jego twórcy dołączają oparte na rzeczywistości, ale zanonimizowane, zestawy danych dla przykładowych projektów. Jest ich 42 - 6 małych problemów i 36 pełnowymiarowych, które posiadają do:

* 200 zadań,
* 150 relacji pomiędzy zadaniami,
* 15 umiejętności na różnych poziomach,
* 40 zasobów,

Jest to problem NP-trudny, co oznacza że nie może zostać on całkowicie rozwiązany w czasie wielomianowym, w związku z czym przegląd zupełny rozwiązań w celu wybrania tego najlepszego nie jest możliwy. W związku z tym do znalezienia akceptowalnych rozwiązań są stosowane różne metaheurystyki. Różnią się one od algorytmów tym, że w przeciwieństwie do nich, nie każdy ich przebieg musi zwrócić to samo stałe rozwiązanie. Nie zwracają one zawsze optymalnego rozwiązania, tylko najlepsze jakie wygenerują, ale dzięki temu czas ich działania zamyka się w rozsądnym przedziale.

Aby podejść do tego problemu należy zdefiniować sobie jakąś funkcję oceny. Możliwe są trzy podejścia: skupienie się na minimalizacji czasu, kosztu, lub pewnej kombinacji tych dwóch naraz. Minimalizacja kosztu jest trywialna – można po prostu wybierać tylko zasoby z najmniejszym kosztem. W związku z tym w swojej pracy zdecydowałem się skupić na minimalizacji czasu. Łączenie ich jest także proste do wykonania, ale zaciemnia wyniki, które są bardziej intuicyjne z przypadku użycia samego czasu.

Kolejną rzeczą jest ustalenie reprezentacji problemu. Taką najprostszą jaka pierwsza przychodzi do głowy były by dwie listy - przydzielenie do każdego zadania danego zasobu i czasu rozpoczęcia. Pozwala to jednak niestety na możliwość wystąpienia konfliktów i powstania nieprawidłowych rozwiązań, gdyż jedno zadanie może zacząć się wykonywać zanim drugie się zakończy. Aby tego uniknąć, zdecydowałem się na zamianę listy czasów rozpoczęcia na listę priorytetów wykonywania zadań. Dzięki temu wciąż jest pełna kontrola na zarówno czasem rozpoczęcia, jak i przypisaniem zasobu do zadania, ale nie mogą wystąpić konflikty. Warto także zauważyć, że redukuje to problem do dwóch połączonych problemów naraz – przypisania, oraz komiwojażera – co pozwala wykorzystywać przykładowo opracowane już wcześniej dla nich operatory genetyczne.

W związku z tym aby obliczyć wartość funkcji oceny dla mojej reprezentacji problemu, należy przejść po kolei przez listę priorytetów zadań, do każdego z nich obliczyć czas zakończenia wykonania się tego zadania, biorąc pod uwagę kiedy dany zasób będzie wolny. Ostatecznym wynikiem funkcji oceny będzie ostatni czas zakończenia zadania, który to jednocześnie jest czasem zakończenia całego projektu.

Niektóre rozwiązania skupiają się tylko na jednej z tych list, a drugą uzupełniają pewną ustaloną przez nich heurystyką. Zmniejsza to znacznie przestrzeń rozwiązań, co pozwala na szybsze wygenerowanie rozwiązania, ale może także nie pozwolić na wybranie tak dobrego, dlatego nie zdecydowałem się na tą opcję.

Ważnym elementem który używany jest w wielu metaheurystykach jest kwestia generowania sąsiada. Powinny to być osobniki które różnią się w najmniejszy możliwy sposób od obecnego osobnika. W związku z tym dla tego problemu zdefiniowałem dwa sposoby generacji sąsiada:

* Zamiana dwóch sąsiadujących zadań miejscami w kolejce zadań
* Zmiana wykonującego zadanie na innego który wciąż posiada wymaganą umiejętność do jego wykonania