

## 희소성 알고리즘

	<p><b>희소성 알고리즘 정의 및 목표</b></p> <ul style="list-style-type: none"><li>• 하드 커런시를 3종 중 2종은 충분하고 1종만 부족한 상태 유발<ul style="list-style-type: none"><li>◦ 이 상태를 거의 성공(Near-Miss)라고 명명</li></ul></li><li>• 단순히 1종이 부족한 것이 목표가 아님<ul style="list-style-type: none"><li>◦ 1종이 많이 부족한 상태, 적게 부족한 상태를 오가야 함</li><li>◦ 이를 통해 좌절감을 완화하고 거의 다 왔다는 느낌을 줘야 함</li></ul></li></ul>
	<p><b>결핍 대상/비결핍 대상</b></p> <ul style="list-style-type: none"><li>• 3종 중 적게 획득할 대상은 결핍 대상이라고 명명</li><li>• 나머지 2종은 비결핍 대상이라고 명명</li></ul>
	<p><b>결핍 대상 선정 조건</b></p> <ul style="list-style-type: none"><li>• 업그레이드 필요량 대비 현재 보유량이 충분히 적어야 함<ul style="list-style-type: none"><li>◦ 3종의 도구의 진행도가 모두 0.33 이하여야 함<ul style="list-style-type: none"><li>▪ 진행도란 보유량/필요량을 0~1로 정규화한 값임</li></ul></li></ul></li></ul> <p><b>현재 선정 조건의 한계점</b></p> <ul style="list-style-type: none"><li>• 결국에는 결핍 대상이 바뀌기 힘들 정도의 불균형이 발생할 것임</li><li>• 하지만, 아래 요건을 만족시키면서 결핍 대상을 변경하는 것이 어려움</li><li>• 따라서 고착화 방지 로직은 Todo로 남김</li></ul>

	<ul style="list-style-type: none"> <li>◦ 불균형이 크면 결핍 대상 변경 시 2종 충분, 1종 결핍 상태를 유지하기 까다로움</li> <li>◦ 시스템이 의도적으로 1종 결핍을 유도한다는 걸 알기 힘들어야 함 (불쾌함을 느낄 정도로 노골적이면 안됨)</li> <li>◦ 기차 집중 플레이 대응, 레벨 대비 창고 크기 등으로 인해 불균형을 의도적으로 고착화시킬 수 있음</li> </ul>
	<p><b>결핍 대상 선정 시점</b></p> <ul style="list-style-type: none"> <li>• 시점1: 창고 업그레이드 시 <ul style="list-style-type: none"> <li>◦ 정확히는 창고 업그레이드 시점이 아니라 업그레이드에 필요한 자원을 확보하는 시점임 <ul style="list-style-type: none"> <li>▪ 창고 업글을 일부러 안 하는 경우 대응용</li> </ul> </li> </ul> </li> <li>• 시점2: 창고 확장 도구 판매 시 <ul style="list-style-type: none"> <li>◦ 도구 판매로 선정 조건을 만족하는 순간 (3종 진행도 0.33 이하)</li> <li>◦ 다만, 어뷰징 등을 방지하기 위해 1번 발동하면 쿨타임 존재 <ul style="list-style-type: none"> <li>▪ 시점1이 도래해야 쿨타임 리셋됨</li> </ul> </li> </ul> </li> <li>• 확률 적용 <ul style="list-style-type: none"> <li>◦ 시점이 도달했다고 무조건 결핍 대상 변경을 시도하지 않음</li> <li>◦ 시스템 파훼에 대응하기 위해 확률적으로 결핍 대상 변경 시도</li> </ul> </li> <li>• 대상 리스트 <ul style="list-style-type: none"> <li>◦ 위 확률을 뚫었다고 무조건 결핍 대상이 변경되지 않음</li> <li>◦ 결핍/비결핍 대상 모두 후보로 넣고 결핍 대상 선정</li> </ul> </li> </ul>
	<p><b>결핍 대상 선정 방법</b></p> <ul style="list-style-type: none"> <li>• 공정성보다는 예측 불가능성 우선</li> <li>• 수식: 각 도구의 진행도를 가중치로 사용 <ul style="list-style-type: none"> <li>◦ 0.3 / 0.2 / 0.1을 그대로 가중치로 사용하여 결핍 대상 고착화 완화</li> </ul> </li> </ul>

	<p><b>드랍률 조정</b></p> <ul style="list-style-type: none"> <li>결핍 대상 대비 각각의 비결핍 대상 진행도 격차로 드랍률 설정</li> </ul>
	<p><b>진행도 격차</b></p> <ul style="list-style-type: none"> <li>진행도: 위에 언급한대로 보유량/업글 필요량을 0~1로 정규화한 값</li> <li>진행도 격차: 비결핍 대상 진행도 - 결핍 대상 진행도 <ul style="list-style-type: none"> <li>진행도가 0.3 / 0.2 / 0.1이고 0.2가 결핍 대상이라고 가정</li> <li>비결핍 대상의 진행도 격차는 0.1과 -0.1임</li> </ul> </li> <li>비결핍 도구별 계산 <ul style="list-style-type: none"> <li>비결핍 대상별로 격차를 계산하고 해당 값에 따라 각자 가중치가 변경되는 형태</li> </ul> </li> </ul>
	<p><b>진행도 격차에 따른 드랍률 변화 그래프</b></p> <ul style="list-style-type: none"> <li>결핍 격차가 음수면 비결핍 대상의 드랍률이 매우 높아짐</li> <li>결핍 격차가 0이면 기본적인 불균형 상태</li> <li>결핍 격차가 0~0.5이면 불균형 심화 상태로 드랍률 격차가 점점 커짐</li> <li>결핍 격차가 0.5~0.75면 불균형 완화 상태로 드랍률 격차가 점점 감소</li> <li>결핍 격차가 0.75~1이면 불균형 역전 상태로 결핍 대상의 드랍률이 오히려 높아짐 <ul style="list-style-type: none"> <li>불균형 역전 상태가 없으면 불균형이 점점 심해지기만 함</li> <li>따라서 이론적으로 창고 확장 도구 2종만으로 창고가 가득 찰 수도 있음</li> </ul> </li> <li>의도한 사항 <ul style="list-style-type: none"> <li>불균형 심화, 완화를 반복하여 좌점갈을 완화하고 거의 성공 상태를 지속적으로 유발</li> </ul> </li> <li>우려점 <ul style="list-style-type: none"> <li>구간별 패턴이 고정되어 있어 시스템이 쉽게 파악될 수 있음</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>◦ 결핍 대상이 업글 필요량에 가까워 질 수록 드랍률이 낮아져야 ‘거의 성공’ 상태가 효과적으로 작동함 <ul style="list-style-type: none"> <li>▪ 불균형 심화/완화/역전/심화 상태로 가야 하는 걸까?</li> </ul> </li> </ul>
	<p><b>결핍 격차 - 0인 상태</b></p> <ul style="list-style-type: none"> <li>• 기본적인 불균형 상태</li> <li>• 도구 3종의 진행도가 모두 낮을 때 불균형을 빠르게 생성해야 하기 때문에 기본적인 불균형 지수 존재 <ul style="list-style-type: none"> <li>◦ 이때 불균형 지수 값을 U라고 명명하고 불균형의 기준으로 삼음 <ul style="list-style-type: none"> <li>▪ 현재 예시에서는 U가 1.5라고 가정</li> <li>▪ 비결핍 2종의 결핍 격차가 둘다 0이고 U가 1.5면 드랍률 가중치를 3:3:2로 설정</li> </ul> </li> </ul> </li> </ul>
	<p><b>결핍 격차 - 음수인 상태</b></p> <ul style="list-style-type: none"> <li>• 비결핍 대상이 오히려 적기 때문에 드랍률을 매우 높임 <ul style="list-style-type: none"> <li>◦ 비결핍 대상의 드랍률 가중치를 <math>U + \text{결핍 격차} * -1 * M</math>으로 설정</li> <li>◦ M의 값은 1 이상임</li> </ul> </li> </ul>
	<p><b>결핍 격차 - 불균형 심화 상태</b></p> <ul style="list-style-type: none"> <li>• 결핍 격차가 0~0.5인 도구 대상으로 수행</li> <li>• 불균형을 점점 가속화하는 단계</li> <li>• 비결핍 대상의 드랍률 가중치를 <math>U + \text{결핍 격차} * S</math>로 설정</li> <li>• S의 값은 1 이상임 <ul style="list-style-type: none"> <li>◦ M보다는 낮은 값으로 설정할 예정</li> </ul> </li> </ul>
	<p><b>결핍 격차 - 불균형 완화 상태</b></p> <ul style="list-style-type: none"> <li>• 결핍 격차가 0.5~0.75인 도구 대상으로 수행</li> <li>• 불균형을 완화하는 단계</li> <li>• 비결핍 대상의 드랍률 가중치를 <math>U + (\text{결핍 격차} - 0.5) * -R</math>로 설정</li> </ul>

	<ul style="list-style-type: none"> <li>• R의 값은 <math>2 \times S</math>임 <ul style="list-style-type: none"> <li>◦ 결핍 격차가 0.75일때 드랍률이 균등해짐</li> </ul> </li> </ul>
	<p><b>결핍 격차 - 불균형 역전 상태</b></p> <ul style="list-style-type: none"> <li>• 결핍 격차가 0.75~1인 도구 대상으로 수행</li> <li>• 결핍 대상의 드랍률이 오히려 높은 상태</li> <li>• 비결핍 대상의 드랍률 가중치를 <math>U + (\text{결핍 격차} - 0.5) \times -R</math>로 설정</li> </ul>
	<p><b>TODO</b></p> <ul style="list-style-type: none"> <li>• 수식 고도화: 현재는 선형 함수와 사칙 연산만 사용 <ul style="list-style-type: none"> <li>◦ 인력 총원이나 게임 안정화 이후 고도화될 수 있음</li> </ul> </li> </ul>
	<p><b>기획 관점에서 주의 사항</b></p> <ul style="list-style-type: none"> <li>• 도구 교환소나 유저 간 거래 등 특정 도구를 확정적으로 얻을 시스템 및 컨텐츠는 극히 제한적이어야 함</li> </ul>